НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено:

Завідувач кафедри

_____ Олександр Коваль

«___»_____2020 р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інформаційні технології моніторингу довкілля»

спеціальності 122 «Комп'ютерні науки та інформаційні технології»

на тему: «Моніторинг надзвичайних ситуацій техногенного характеру»

Виконав (-ла):
студент (-ка) IV курсу, групи ТМ-62
Коломоєць Іван Михайлович _____

Керівник:
Ст-викл., к.ф.-м.н.
Шульженко Олег Феодосійович _____
_____

Рецензент:
Головний інженер науково-інженерного центру "Водоекологія"
к.т.н. Писарук Віктор Іванович _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2020 року

Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем
Рівень вищої освіти перший рівень
Напрям підготовки 122 Комп'ютерні науки та інформаційні технології
Спеціалізація Інформаційні технології моніторингу довкілля

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Олександр Коваль
(підпис)
"___"_____2020р.

ЗАВДАННЯ
на дипломну роботу студенту

_____Коломойцю Івану Михайловичу__
(прізвище, ім'я, по батькові)

1. Тема роботи «Система обліку надзвичайних подій техногенного характеру»
_____
керівник роботи____ст-викл._ к.ф.-м.н. Шульженко Олег Феодосійович___
(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від "25" травня 2020р. № **1168-с**

2. Строк подання студентом роботи _____
3. Вихідні дані до роботи _____
_____

_____

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)____проаналізувати існуючу систему моніторингу надзвичайних ситуацій техногенного характеру, спроектувати архітектуру веб-додатку з моніторингу надзвичайних ситуацій техногенного характеру, розробити програмне забезпечення, розробити інтерфейс користувача_____

5. Перелік ілюстративного матеріалу

1. Задача створення веб-додатку з моніторингу надзвичайних ситуацій воєнного характеру_2.Характеристика надзвичайних ситуацій_3.Моніторинг надзвичайних ситуацій_4.Аналіз проблеми створення веб-додатку з моніторингу надзвичайних ситуацій воєнного характеру__5.Засоби розробки_6.Опис реалізації_7.Робота користувача з системою_

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання "___"_____20___ р.

АННОТАЦИЯ

Целью работы являлась разработка системы учёта данных о техногенных катастрофах. Приложение поддерживает отображение местоположения чрезвычайных событий на Украине, работу с информацией касательно техногенных катастроф, её хранение и отображение.

Административная панель позволяет редактировать данные включая выбор расположение на карте. Система также поддерживает калькулятор расчёта убытков. Приложение создано в соответствии современным подходам разработки программного обеспечения.

Записка содержит 56 страниц, 12 изображений и 11 источников.

ЗМІСТ

ВСТУП

Життєдіяльність людини спрямована на полегшення та покращення власного існування, в тому числі завдяки техногенному прогресу.

На сьогоднішній день техногенний прогрес невтомно продовжує розвиток, за рахунок якого з’являється різноманітна інноваційна техніка, збільшуються технічні можливості людства, автоматизується значна частина людської праці. Спроби рухати прогрес в даній області, на жаль, періодично призводять до різноманітних небажаних техногенних катастроф.

Кількість ситуацій, в яких кожна катастрофа, пов’язана з людською діяльністю, значно зросла з кінця 1950-х років. Сьогодні такі явища вважаються надзвичайно небезпечними, непередбачуваними, широкими з точки зору людських жертв і їх дуже важко усунути.

Що таке техногенна катастрофа, можна описати так - це все

Порушення нормальних умов життя та діяльності окремих людей окремо

Територія чи об’єкт на ній чи водойма, спричинені

Аварія, катастрофа, стихійне лихо чи інше небезпечне

Подія, що включає епідемію, епідемію, епіфітоз, пожежу,

Що призвело (може призвести) до неможливості життя населення

У полі чи об'єкті, де є господарська діяльність,

Значні смерті та / або значні втрати; [2]

Найважливішими ознаками таких аварій є їх здивування та аварія.Технологічні катастрофи звичайно протиставляють природним, проте і це вимагає уточнення. Всі лиха в кінцевому рахунку є наслідками тих чи інших людських дій або відсутності таких. Катастрофа будь-якого

походження - це фізична подія в громадському контексті. Технологічні (техногенні) катастрофи також в своїй основі мають соціальні причини, оскільки технічні системи конструюються, виготовляються і управляються людьми і забезпечують досягнення тих чи інших соціально значущих цілей. Енергетичні, ядерні, інфраструктурні, транспортні, екологічні та космічні аварії і катастрофи, в кінцевому рахунку, викликаються неузгодженістю взаємодії елементів складних систем, в створенні і функціонуванні яких задіяні як люди, так і ті чи інші елементи створених ними технологій. У цьому типі катастроф у міру розвитку техніки все більшу роль починає грати людський фактор, який проявляється в інженерних прорахунках, помилках персоналу, неефективної допомоги рятувальних служб. Зростання розмірів і потужності технічних систем підвищує ризик людських, матеріальних і екологічних втрат - така плата за технологічний прогрес.

Виникнення будь-якої надзвичайної ситуації, в тому числі і техногенної катастрофи, викликається поєднанням дій об'єктивних і суб'єктивних чинників, що створюють причинний ряд подій. Безпосередніми причинами техногенних катастроф можуть бути зовнішні по відношенню до інженерної системі впливу (стихійні лиха, військово-диверсійні акції і т.д.), умови та обставини, пов'язані безпосередньо з даною системою, в тому числі технічні несправності, а також людські помилки. Останнім, згідно зі статистикою і думку фахівців, належить головна роль у виникненні техногенних катастроф. За оцінкою експертів, людські помилки обумовлюють 45% екстремальних ситуацій на АЕС, 60% авіакатастроф і 80% катастроф на морі.

Все частіше трапляються катастрофічні аварії із руйнуванням об'єктів та серйозними наслідками для навколишнього середовища. Аналіз таких ситуацій показує, що незалежно від виробництва, у переважній більшості випадків вони мають однакові стадії розвитку.

3. Наслідки та правила поведінки під час технологічних катастроф.

Техногенна катастрофа відноситься до надзвичайних ситуацій із усіма соціальними, правовими та економічними наслідками, які вони мають у суспільстві, глобального характеру.

Аварії та катастрофи не мають національних кордонів, вони спричиняють загибель людей, а в свою чергу створюють соціальну та політичну напругу (наприклад, Чорнобильська аварія [1]). Тисячі потенційно небезпечних предметів що використовуються на всіх континентах у таких кількостях радіоактивних, вибухонебезпечних та токсичних речовин, що в надзвичайних ситуаціях можуть завдати незворотної втрати навколишньому середовищу або навіть знищити життя на Землі.

У разі великих аварій та катастроф організація ліквідаційних робіт враховує ситуацію після аварії чи катастрофи, ступінь руйнування та пошкодження будівель та споруд, технологічного обладнання, агрегатів, характер аварій в мережах обслуговування та пожеж, характеристики об'єктів. умови. Робота з ліквідації аварії та її катастрофічних наслідків починається різко: людей потрібно якнайшвидше врятувати від уламків підвалу, надавати екстрену медичну допомогу, а також запобігати іншим катастрофічним наслідкам загибелі та втрати великої кількості необхідних матеріалів .

Велику небезпеку становлять техногенні катастрофи, які виникають внаслідок порушення технологічного процесу або раптового виходу з ладу машин, механізмів і технічних пристроїв під час їх експлуатації. До техногенних катастроф відносяться різні аварії на промислових і енергетичних об'єктах, а також на транспорті, розтікання по поверхні грунту і води токсичних рідин і нафтопродуктів та ін.

Великі аварії і катастрофи техногенного характеру в останні десятиліття мали істотний вплив на життя і здоров'я планети, середовище її проживання.

Антропогенні катастрофи визначаються людським фактором, тому проводяться роботи для їх запобігання: перевірка зносу, дисциплінованості та професіоналізму працівників. Оскільки можливості техногенних катастроф неможливо запобігти, необхідно вчасно вжити заходів щодо запобігання можливого її виникнення, планувати її заселення, евакуацію населення з постраждалого району та допомогу жертвам та жертвам у зоні стихійного лиха.

Аварії та стихійні лиха - поширене явище в нашій країні, кожне з яких має свої особливості, характер пошкоджень, масштаби та масштаби руйнування, ступінь стихійних лих та людських втрат. Знання причин та надзвичайних ситуацій техногенного характеру дозволяє своєчасно вживати гарантій при розумній поведінці населення, щоб значно зменшити всі види шкоди. Все населення має бути готовим до надзвичайних ситуацій і знати першу допомогу жертвам.

Записка містить 5 розділів.

У першому розділі описується постановка задачі з дослідження катастрофи техногенного характеру

У другому розділі описується техногенні катастрофи та їх особливості.

У третьому розділі описані засоби створення системи обліку.

У четвертому розділі описані деталі реалізації програмного продукту.

У п'ятому розділі описано роботу користувача з системою.

1. ПОСТАНОВКА ЗАДАЧІ З ДОСЛІДЖЕННЯ КАТАСТРОФИ ТЕХНОГЕННОГО ХАРАКТЕРУ

1.1. Постановка задачі створення системи обліку надзвичайних ситуацій техногенного характеру

З розвитком сучасного світу та технологій важливо ефективно систематизувати та аналізувати дані. Тож ефективність роботи з даними напряму залежить від їхнього представлення та зручності використання, редагування, видалення тощо.

Інтуїтивна зрозумілість для користувача інтерфейсу, яка не йде в розріз з функціоналом є головним завданням даної роботи. Інформація разом з цим не повинна мати надлишковий характер. Завдання маніпулювання даними зосереджувалася на агрегації інформації з різних джерел. Необхідно було створити систему обліку питань надзвичайних ситуацій техногенного характеру, яка дозволила б зображати події інтерактивним чином на мапі України. В тому числі, програмне забезпечення повинно вирішувати проблему зручного відображення інформації по конкретному події, а також списку всіх ситуацій. Система повинна враховувати час подій, і угрупованням за даними критеріями.

З метою дослідження катастроф техногенного характеру було запропоновано розробити програмну систему, яка продемонструє кількість надзвичайних ситуацій в регіонах.

1.2. Постановка задачі пошуку даних надзвичайних ситуацій техногенного характеру

У даній роботі використовуються офіційні дані з сайту Верховної Ради України щодо техногенних катастроф з метою інтерактивного відображення і додавання шляхом вибору місця події і записи відповідних даних. Основними вхідними даними системи була загальна інформація про подію із зазначенням їх назви населеного пункту, дати, опису, кількості постраждалих, загиблих, фінансові витрати на ліквідацію того чи іншого події.

.

1.3. Постановка задачі загальної реалізації системи обліку надзвичайних ситуацій техногенного характеру

Отримана система має буди легко розширюванною, відповідати сучасним підходам розробки програмного забезпечення, зручною у використанні цільовими користувачами. Необхідно відображати та зберігати лише необхідні для опису тієї чи іншої катастрофи даними, уникаючи великої кількості непотрібної інформації. Невід’ємною частиною створення має бути також забезпечення доступності користувачами, мінімальним системним вимогам. Розробка зрозумілого інтерфейсу.

1.4 Потенційні користувачі

Беручи до уваги функціональність додатку, а саме зберігання та обліку інформації про надзвичайні ситуації різних характерів зі зручним та інтуїтивно зрозумілим інтерфейсом можна зробити висновок що системою можуть цікавитися не лише спеціалісти, а й будь-які зацікавлені особи.

2. ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

2.1. Техногенні катастрофи та їх класифікація

Надзвичайна ситуація, спричинена техногенним сприйняттям - ситуація, коли внаслідок надзвичайних ситуацій, спричинених людиною на місці, певна територія чи акваторія порушує нормальні умови життя та діяльності людей, існує загроза їх життю та здоров'ю, майнові збитки, економіка і навколишнє середовище. Середа.

Існують надзвичайні ситуації техногенного характеру відповідно до їх походження та характеру основних впливових факторів надзвичайних ситуацій.

Джерело надзвичайних ситуацій техногенних ситуацій - небезпечна техногенна аварія, що спричинила техногенну надзвичайну ситуацію на об'єкті, ділянці чи акваторії.

До небезпечних техногенних аварій можна віднести аварії на виробничих або транспортних об'єктах, пожежі, вибухи або різні види викидів енергії.

Виробничі аварії і катастрофи - раптова зупинка або порушення процесу виробництва, що приводить до виникнення пожеж, вибухів, забруднення атмосфери, знищення матеріальних цінностей, супроводжувані ураженням або загибеллю людей [2].

Стрибкоподібні зміни, що виникають у вигляді раптовій відповіді системи на плавну зміну зовнішніх умов, і є катастрофи.

Техногенна катастрофа - це серйозна аварія, яка призводить до масових інцидентів і навіть екологічних катастроф. Однією з особливостей техногенної катастрофи є збіг. Зазвичай техногенні чинять опір стихійним лихам. Однак, як техногенні катастрофи можуть спричинити паніку, крах транспорту.

Зростання виробничих аварій і катастроф, стихійних лих останніх років створює надзвичайні ситуації (НС) з тяжкими наслідками для життя людей і погіршує екологічну обстановку.

Класифікація аварій і катастроф в залежності від причин їх виникнення.

Аварії і катастрофи за характером їх прояви поділяють на кілька груп:

-Транспортні аварії (катастрофи) можуть бути двох видів: що відбуваються на виробничих об'єктах, не пов'язаних безпосередньо з переміщенням транспортних засобів (в депо, на станціях, в портах, на аеровокзалах), і трапляються під час їх руху. Для другого виду аварій характерні віддаленість НС від великих населених пунктів, складність доставки туди рятувальних формувань і велика чисельність постраждалих, які потребують термінової медичної допомоги.

-пожежі і вибухи - найпоширеніші НС. Найбільш часто і, як правило, з важкими соціальними та економічними наслідками вони відбуваються на пожежно та вибухонебезпечних об'єктах. Це перш за все промислові підприємства, які використовують у виробничих процесах вибухові і легкозаймаючі речовини, а також залізничний і трубопровідний транспорт, що несе найбільше навантаження по переміщенню пожежно та вибухонебезпечних вантажів.

-аварії з викидом (загрозою викиду) аварійно хімічно небезпечних речовин (АХОВ) - це події, пов'язані з витоком шкідливих хімічних продуктів в процесі їх виробництва, зберігання, переробки ні транспортування.

-аварії з викидом (загрозою викиду) радіоактивних речовин виникають на радіаційно небезпечних об'єктах: атомних станціях, підприємствах з виготовлення та переробки ядерного палива, захоронення радіоактивних відходів і ін.

-аварії з викидом (загрозою викиду) біологічно небезпечних речовин - не часте явище, що пояснюється, мабуть, суворої засекреченими робіт в цій області і в той же час продуманістю заходів щодо попередження виникнення таких НС. Однак, з огляду на тяжкість наслідків у разі попадання біологічно небезпечних речовин в навколишнє середовище, такі аварії найнебезпечніші для населення.

-раптове обвалення будівель, споруд найчастіше відбуваються не самі по собі, а викликаються побічними факторами: великим скупченням людей на обмеженій площі; сильною вібрацією, викликаної проходять залізничними складами або великовантажними автомобілями; надмірним навантаженням на верхні поверхи будинків і т.д.

-аварії на електроенергетичних системах та комунальних системах життєзабезпечення рідко призводять до загибелі людей. Однак вони суттєво ускладнюють життєдіяльність населення (особливо в холодну пору року), можуть стати причиною серйозних порушень і навіть припинення роботи об'єктів промисловості і сільського господарства.

-аварії на промислових очисних спорудах можуть призвести не тільки до різкого негативного впливу на обслуговуючий персонал цих об'єктів і жителів довколишніх населених пунктів, а й до залповим викидам отруйних, токсичних та просто шкідливих речовин в навколишнє середовище.

-гідродинамічні аварії виникають в основному при руйнуванні (прориві) гідротехнічних споруд, найчастіше гребель. Їх наслідки - пошкодження і вихід з ладу гідровузлів, інших споруд, ураження людей, затоплення великих територій [3].

Серед найбільш небезпечних техногенних (технологічних) катастроф слід вказати аварії на енергетичних об'єктах, перш за все на АЕС; далі йдуть хімічні підприємства, що випускають пестициди, гербіциди, мінеральні добрива, пластмаси; транспортні аварії (при перевезенні небезпечних вантажів); нафтові розливи при прориві трубопроводів та ін. Особливе місце в цьому ряду займає руйнування гребель. За своїми наслідками вони можуть бути більш небезпечними, ніж аварії на АЕС. Слід, однак, підкреслити, що радіаційні та хімічні вражаючі фактори, що виникають при аваріях на АЕС та хімічних підприємствах, мають довгостроковим і, що особливо небезпечно, прихованим (латентним) впливом на організм людини, а також чинять негативний вплив на здоров'я майбутніх поколінь.

2.2. Основні причини техногенних катастроф

Виникнення будь-якої надзвичайної ситуації, включаючи техногенну катастрофу, викликане поєднанням дій об'єктивних та суб'єктивних факторів, що створюють ряд причинно-наслідкових подій. Безпосередніми причинами техногенних катастроф можуть бути зовнішні інженерні системи впливу (стихійні лиха, військові диверсії тощо), умови та обставини, безпосередньо пов'язані з цією системою, включаючи технічні несправності, а також помилки людини. . Останній, за статистикою та експертами, відіграє головну роль у виникненні техногенних катастроф. Експерти кажуть, що людські помилки спричиняють 45% екстремальних ситуацій на атомних електростанціях, 60% авіакатастроф та 80% аварій на морі [2].

На жаль, кількість аварій у всіх сферах виробничої діяльності неухильно зростає. Це відбувається у зв'язку з широким використанням нових технологій і матеріалів, нетрадиційних джерел енергії, масовим застосуванням небезпечних речовин в промисловості і сільському господарстві.

Сучасні складні виробництва розроблені з високим ступенем надійності. Однак чим більше виробничих потужностей, тим більше шансів на щорічну аварію на одному з них. Повної аварії немає.

Все більше катастрофічних аварій із руйнуванням об'єктів та важкими наслідками для навколишнього середовища (наприклад, Чорнобиль). Аналіз таких ситуацій показує, що незалежно від виробництва у переважній більшості випадків вони мають однакові стадії розвитку.

На першій з них аварії зазвичай передує виникнення або накопичення дефектів в обладнанні, або відхилень від нормального ведення процесу, які самі по собі не становлять загрози, але створюють для цього передумови. Тому ще можливо запобігання аварії.

На другій стадії відбувається будь-яка ініціює подія, звичайно несподіване. Як правило, в цей період в операторів зазвичай не буває ні часу, ні коштів для ефективних дій.

Власне катастрофа відбувається на третій стадії, як наслідок двох попередніх.

Таким чином, можна виділити основні причини:

- Прорахунки при проектуванні і недостатній рівень безпеки сучасних будівель;
- Неякісне будівництво або відступ від проекту;
- Непродумане розміщення виробництва;
- Порушення вимог технологічного процесу через недостатню підготовку або недисциплінованість і халатність персоналу.
- Відсутність на належному рівні змісту будівель і споруд, обладнання, що не купуються нові верстати і механізми, замість застарілих.
- Падіння виробничої дисципліни. Неуважність, грубі порушення правил експлуатації техніки, транспорту, приладів і обладнання.
- Сучасне виробництво все більше ускладнюється. У його процесі часто застосовуються отруйні і агресивні компоненти. На малих площах концентрується велика кількість енергетичних потужностей.
- Стихійні лиха, в результаті яких виходять з ладу підприємства, що мають у своєму виробництві небезпечні для суспільства шкідливі речовини і т.д.
- Складність технологій, недостатня кваліфікація персоналу, проектно-конструкторські недоробки, низька трудова і технологічна дисципліна;
- Концентрація різних виробництв в промислових зонах без належного вивчення їх взаємовпливу;
- Відмови технічних систем через дефекти виготовлення і порушень режимів експлуатації;
- Високий енергетичний рівень технічних систем;
- Зовнішні негативні впливи на об'єкти енергетики, транспорту та ін.

Більшість сучасних потенційно небезпечних виробництв спроектовані так, що ймовірність великої аварії на них дуже висока і оцінюється величиною ризику 10 і більше. Статистичні дані показують, що понад 60% аварій сталося в результаті помилок обслуговуючого персоналу[4].

Залежно від виду виробництва, аварії і катастрофи на промислових об'єктах і транспорті можуть супроводжуватися вибухами, виходом ОХВ, викидом радіоактивних речовин, виникненням пожеж і т.п.

2.3 Наслідки і правила поведінки при техногенних НС

Техногенна катастрофа відноситься до надзвичайних ситуацій з усіма наслідками, що випливають соціальними, юридичними та економічними наслідками в сфері суспільства, які носять глобальний характер.

Аварії та катастрофи не мають національних кордонів, вони призводять до загибелі людей і, в свою чергу, створюють соціально-політичну напругу (наприклад, аварія на Чорнобильській АЕС). Тисячі потенційно небезпечних предметів використовуються на всіх континентах світу з такими повними радіоактивними, вибухонебезпечними та токсичними речовинами, що в надзвичайній ситуації можуть завдати незворотної шкоди навколишньому середовищу або навіть знищити життя на Землі.

При великих аваріях і катастрофах організація робіт по ліквідації наслідків проводиться з урахуванням обстановки, що склалася після аварії або катастрофи, ступеня руйнування та пошкодження будівель і споруд, технологічного обладнання, агрегатів, характеру аварій на комунально-енергетичних мережах і пожеж, особливостей забудови території об'єкта та інших умов. Роботи по організації ліквідації наслідків аварій і катастроф проводяться в стислі терміни: необхідно швидко врятувати людей, що знаходяться під уламками будівель, в завалених підвалах, і надати їм екстрену медичну допомогу, а також запобігти інші катастрофічні наслідки, пов'язані із загибеллю людей і втратою великої кількості матеріальних цінностей.

Радіаційна аварія

У наш час практично в будь-якій галузі господарства і науки в усі більш зростаючих масштабах використовуються радіоактивні речовини і джерела іонізуючих випромінювань. Особливо високими темпами розвивається ядерна енергетика. Атомна наука і техніка таять в собі величезні можливості, але

разом з тим і велику небезпеку для людей і навколишнього середовища, про що свідчать аварії на АЕС, АПЛ, атомних криголамах, літаках - носіях ядерної зброї, космічних літальних апаратах.

Ядерні матеріали доводиться возити, зберігати, переробляти. Всі ці операції створюють додатковий ризик радіоактивного забруднення місцевості, ураження людей, рослинного і тваринного світу.

Гостро стоїть проблема завезення і поховання на території області відпрацьованого ядерного палива. Наукові дослідження в області ядерної фізики в ЗАТО р Саров мають потенційну можливість аварії, яка може призвести до ураження людей.

Основними вражаючими факторами таких аварій є радіаційний вплив і радіоактивне забруднення. Аварії можуть супроводжуватися вибухами і пожежами [5].

Радіаційний вплив на людину полягає в порушенні життєвих функцій різних органів (головним чином органів кровотворення, нервової системи, шлунково-кишкового тракту) і розвитку променевої хвороби під впливом іонізуючого випромінювання здійснюватиме.

Радіоактивне забруднення викликається впливом альфа-, бета-і гамма іонізуючих випромінювань і обумовлюється виділенням при аварії не прореагували елементів і продуктів поділу ядерної реакції (радіоактивний шлак, пил, осколки ядерного продукту), а також утворенням різних радіоактивних матеріалів і предметів (наприклад, грунту) в результаті їх опромінення.

Попереджувальні заходи.

Уточнити наявність поблизу вашого місця розташування радіаційно-небезпечних об'єктів та отримати, можливо, більш детальну і достовірну інформацію про них. З'ясувати в найближчому територіальному управлінні у справах ГОЧС способи і засоби оповіщення населення при аварії на Вас

зацікавило радіаційно-небезпечному об'єкті та переконатися в справності відповідного обладнання [7].

Вивчити інструкції про порядок Ваших дій у разі радіаційної аварії.

Створити запаси необхідних коштів, призначених для використання в разі аварії (герметизуючих матеріалів, йодних препаратів, продовольства, води і т.д.).

Як діяти при оповіщенні про радіаційної аварії.

Коли ви на вулиці, одразу захистіть дихальну систему шарфом і поспішіть ховатися в приміщенні. Після укриття зніміть верхній одяг та взуття, покладіть їх у поліетиленовий пакет і прийміть душ. Закрийте вікна та двері. Увімкніть телевізор і радіо, щоб отримати докладнішу інформацію про аварію та вказівки органів місцевого самоврядування. Пломбуйте отвори, тріщини на вікнах (дверях) і не зайвим чином наближайтесь до них. Запас води в герметичній тарі. Загорніть відкриті вироби в пластикову упаковку і поставте в холодильник (шафу).

Для захисту органів дихання використовуйте респіратор, ватно-марлеву пов'язку або підручні вироби з тканини, змочені водою для підвищення їх фільтруючих властивостей [5].

При отриманні вказівок через засоби масової інформації проведіть йодну профілактику, приймаючи протягом 7 днів по одній таблетці (0,125 г) йодистого калію, а для дітей до 2-х років - j частину таблетки (0,04 г). При відсутності йодистого калію використовуйте йодистий розчин: три-п'ять крапель 5% розчину йоду на склянку води, дітям до 2-х років - одну-дві краплі.

Як діяти на радіоактивно забрудненій місцевості.

Для попередження або послаблення впливу на організм радіоактивних речовин:

- виходьте з приміщення тільки в разі потреби і на короткий час, використовуючи при цьому респіратор, плащ, гумові чоботи і рукавички;

- на відкритій місцевості не роздягайтеся, не сідайте на землю і не курите, виключіть купання в відкритих водоймах і збір лісових ягід, грибів;

- територію біля будинку періодично зволожуйте, а в приміщенні щодня проводите ретельне вологе прибирання із застосуванням миючих засобів;

- перед входом в приміщення вимийте взуття, витрусити і почистіть вологою щіткою плаща

- воду вживайте тільки з перевірених джерел, а продукти харчування - придбані в магазинах;

- ретельно мийте перед їжею руки і полощіть рот 0,5% -м розчином питної соди,

Дотримання цих рекомендацій допоможе уникнути променевої хвороби.

Хімічна аварія.

Це порушення технологічних процесів на виробництві, пошкодження трубопроводів, ємностей, сховищ, транспортних засобів, що приводить до викиду аварійних хімічно небезпечних речовин (АХОВ) в атмосферу в кількостях, які становлять небезпеку для життя і здоров'я людей, функціонування біосфери.

Хімічно небезпечний об'єкт (ХНО) - підприємство, в провадженні якого застосовуються аварійно-хімічно небезпечні речовини (АХОВ) і при аварії або руйнуванні якого можуть відбутися масові ураження людей, тварин і рослин хімічно небезпечними речовинами.

Аварійні викиди ахова можуть статися при пошкодженнях і руйнування ємностей при зберіганні, транспортуванні або переробці. Крім того, деякі нетоксичні речовини в певних умовах (вибух, пожежа) в результаті хімічної

аварії можуть утворити ахова. У разі аварії відбувається не тільки зараження приземного шару атмосфери, а й зараження водних джерел, продуктів харчування, грунту.

Небезпека хімічної аварії для людей і тварин полягає в порушенні нормальної життєдіяльності організму і можливості віддалених генетичних наслідків, а при певних обставинах - в летальному результаті при попаданні ахова в організм через органи дихання, шкіру, слизові оболонки, рани і разом з їжею.

Попереджувальні заходи.

Дізнайтеся, чи поблизу вашого місця проживання чи роботи є хімічно небезпечний об’єкт. Якщо це так, будьте ознайомлені з особливостями, унікальними особливостями та потенційними ризиками, які існують у об'єкті. Пригадайте характерні риси тривожного сигналу про аварію "Увага - всі!" (Вихлюють сирени та переривчасті звукові сигнали фабрик), процедура її отримання, правила герметизації приміщення, захист їжі та води. Підготуйте та зберігайте бавовняні марлеві пов’язки, доступні вам та вашій родині, а також пам’ятайте про дії населення у разі аварії на хімічно небезпечному об’єкті. По можливості купуйте протигази з ящиками, які захищають від відповідних видів захисту.

Як діяти при хімічної аварії [8].

При сигналі "Увага - ВСІМ!" включите радіоприймач і телевізор для отримання достовірної інформації про аварію і рекомендованих діях.

Закрийте вікна, вимкніть електропобутові прилади і газ. Одягніть гумові чоботи, плащ, візьміть документи, необхідні теплі речі, 3-х добовий запас псуються продуктів, повідомте сусідів і швидко, але без паніки виходьте із зони можливого зараження перпендикулярно напрямку вітру, на відстань не менше 1,5 км від попереднього місця перебування . Для захисту органів дихання використовуйте протигаз, а при його відсутності - ватно-марлеву пов'язку або підручні вироби з тканини, змочені у воді, 2-5% -ному розчині харчової соди

(для захисту від хлору), 2% -ому розчині лимонної або оцтової кислоти (для захисту від аміаку).

При неможливості покинути зону зараження щільно закрийте двері, вікна, вентиляційні отвори і димоходи. Наявні в них щілини заклейте папером або скотчем. Чи не укривайтеся на перших поверхах будівель, в підвалах і напівпідвалах.

При аваріях на залізничних і автомобільних магістралях, пов'язаних з транспортуванням ахова, небезпечна зона встановлюється в радіусі 200 м. Від місця аварії. Наближатися до цієї зони і входити в неї категорично заборонено.

Як діяти після хімічної аварії.

При підозрі на АНОV будь-які вправи слід виключити, вживати з великою кількістю рідини (молока, чаю) та негайно звернутися до лікаря. Вхід до будівлі дозволяється лише після перевірки безпеки. Якщо ви знаходитесь під прямим впливом безпеки, прийміть душ з першої ж можливості. Прати забруднений одяг, а якщо ви не можете прати - викиньте. Добре очистіть приміщення. Уникайте пиття водопровідної води, фруктів та овочів із саду, м’яса тварин та птиці, забитої після аварії, до офіційного висновку щодо їх безпеки.

Пожежі і вибухи.

Найбільш поширеними джерелами виникнення надзвичайних ситуацій техногенного характеру є пожежі і вибухи, які відбуваються:

- на промислових об'єктах;

- на об'єктах видобутку, зберігання і переробки легкозаймистих, горючих та вибухових речовин;

- на транспорті;

- в шахтах, гірських виробках, метрополітенах;

- в будівлях і спорудах житлового, соціально-побутового та культурного призначення.

Пожежа - це вийшов з-під контролю процес горіння, що знищує матеріальні цінності і створює загрозу життю і здоров'ю людей. У Росії кожні 4-5 хвилин спалахує пожежа і щорічно гине від пожеж близько 12 тисяч чоловік.

Основними небезпечними факторами пожежі є теплове випромінювання, висока температура, отруйну дію диму (продуктів згоряння: окису вуглецю та ін.) І зниження видимості при задимлення. Критичними значеннями параметрів для людини, при тривалому впливі зазначених значень небезпечних факторів пожежі, є: температура, щільність теплового випромінювання, концентрація окису вуглецю, видимість в зоні задимлення.

Вибух - це горіння, яке супроводжується виділенням великої кількості енергії за обмежений час. Вибух призводить до утворення і поширення надвукової швидкості доменної хвилі (при надлишковому тиску понад 5 кПа), що має механічний вплив на навколишні об'єкти.

Головними помітними причинами вибуху є повітряна ударна хвиля та розщеплення полів, створених літаючими сміттями різних предметів, технологічного обладнання та вибухових речовин.

Попереджувальні заходи.

У число попереджувальних заходів можуть бути включені заходи, спрямовані на усунення причин, які можуть призвести до виникнення пожежі (вибух), на обмеження (локалізацію) поширення пожеж, створення умов для евакуації людей і майна під час пожежі, своєчасне виявлення пожежі та оповіщення про нього, гасіння пожежі , підтримання сил ліквідації пожеж в постійній готовності.

Дотримання технологічних режимів виробництва, утримання обладнання, особливо енергетичних мереж, в справному стані дозволяє, в більшості випадків, виключити причину загоряння.

Своєчасне виявлення пожежі може досягатися оснащенням виробничих і побутових приміщень системами автоматичної пожежної сигналізації або, в окремих випадках, за допомогою організаційних заходів.

Первісне гасіння пожежі (до прибуття викликаних сил) успішно проводиться на тих об'єктах, які оснащені автоматичними установками гасіння пожежі.

3. ЗАСОБИ РЕАЛІЗАЦІЇ ПРОГРАМНОЇ СИСТЕМИ

Під час розробки веб додатку для система обліку надзвичайних ситуацій техногенного характеру можна виділити такі основні етапи:

- Створення концептуальної моделі бази даних
- Розробка веб додатку на основі моделі

3.1 Мова програмування C#

C# – сучасна об'єктно-орієнтована мова програмування. C # відноситься до широко відомому сімейству мов C, добре знайома кожному, хто працював з C, C++, Java.

C# є об'єктно-орієнтованою мовою, але підтримує також і компонентно-орієнтоване програмування. Розробка сучасних додатків все більше тяжіє до створення програмних компонентів у формі автономних і самоописуваних пакетів, що реалізують окремі функціональні можливості. Головна особливість таких компонентів в тому, що вони являють собою модель програмування з властивостями, методами і подіями. У них є атрибути, що надають декларативні відомості про компоненті. Вони включають в себе власну документацію. C# надає мовні конструкції, безпосередньо підтримують таку концепцію роботи. Завдяки цьому C# підходить для створення і застосування програмних компонентів [9].

Ось лише кілька функцій мови C #, які забезпечують надійність та стабільність програми. Збір сміття автоматично звільняє зайняту пам’ять недосяжними об’єктами. Обробка винятків забезпечує вбудований підхід та розширення для виявлення та відновлення помилок. Структура безпечних типів мови не дозволяє читати нечитабельні змінні, індексувати поза межами чи неперевірені типи привидів.

У C# існує єдина система типів. Всі типи C#, включаючи типи-примітиви, такі як int і double, успадковують від одного кореневого типу object. Таким

чином, всі типи використовують загальний набір операцій, і значення будь-якого типу можна зберігати, передавати і обробляти схожим чином. Крім того, C # підтримує призначені для користувача посилальні типи і типи значень, дозволяючи як динамічно виділяти пам'ять для об'єктів, так і зберігати спрощені структури в стеці.

Для забезпечення сумісності програм і бібліотек C # з подальшим розвитком велика увага приділялася контролю версій у розробці C #. Багато мов програмування ігнорують це питання. В результаті програми на цих мовах частіше порушуються, ніж нам би хотілося, коли випускаються нові версії залежних бібліотек. Питання контролю версій значно вплинули на аспекти розробки C #, такі як окремі віртуальні та непрямі зміни, правила розв’язання перевантажень методами та підтримка явного висловлення членами інтерфейсу.

В останніх версіях C # використовуються інші парадигми програмування. C # включає функції, які підтримують функціональні методи програмування, такі як лямбда-вирази. Інші нові функції підтримують поділ даних та алгоритми, такі як відповідність шаблонів.

3.2 Common Language Runtime (CLR)

.NET Framework пропонує середовище виконання часу, яке називається загальною мовою виконання, яке запускає код і надає послуги, що полегшують процес розробки [10].

Компілятори та інструменти розкривають функціональні можливості загальної мови виконання та дозволяють писати код, який корисний від цього керованого середовища виконання. (рис 3.1) Код, який ви розробляєте за допомогою компілятора мови, який орієнтується на час виконання, називається керованим кодом; йому вигідні такі функції, як інтеграція між мовами, обробка виключень між мовами, покращена безпека, підтримка версій та розгортання, спрощена модель взаємодії компонентів та послуги налагодження та профілювання.

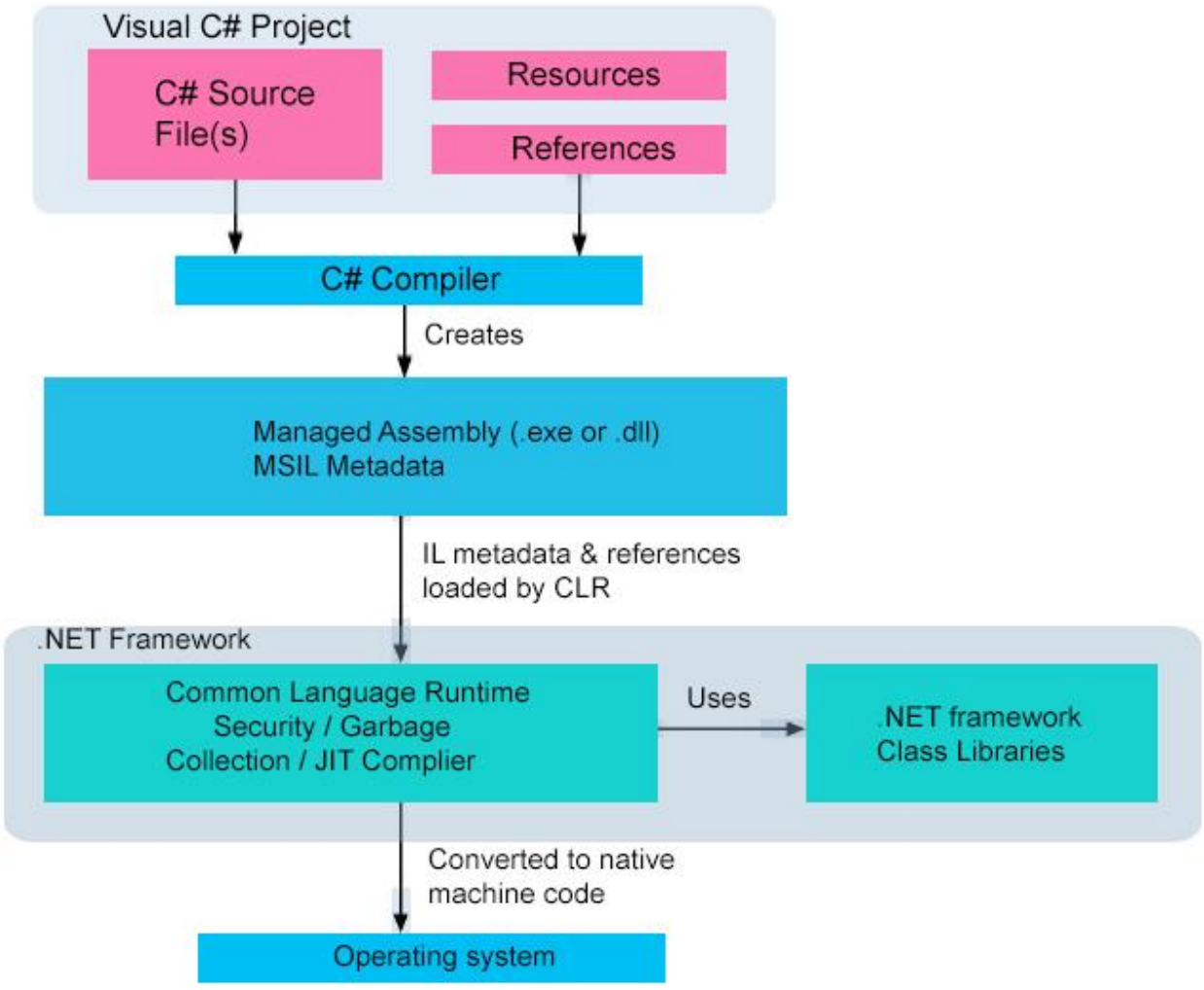


Рисунок 3.1 – Схема роботи CLR.

3.3 .NET Core

Платформа ASP.NET Core [11] представляє технологію від компанії Microsoft, призначену для створення різного роду веб-додатків: від невеликих веб-сайтів до великих веб-порталів і веб-сервісів [11].

З одного боку, ASP.NET Core є продовженням розвитку платформи ASP.NET. Але з іншого боку, це не просто черговий реліз. Вихід ASP.NET Core фактично означає революцію всієї платформи, її якісна зміна.

Розробка над платформою почалася ще в 2014 році. Тоді платформа умовно називалася ASP.NET vNext. У червні 2016 року вийшов перший реліз платформи. А в грудні 2019 року побачила версія ASP.NET Core 3.1, яка власне і буде охоплена в поточному керівництві.

ASP.NET Core тепер повністю є opensource-фреймворком. Всі вихідні файли фреймворку доступні на GitHub [11].

ASP.NET Core може працювати в середовищі міжнародної платформи. NET Core, який можна розгорнути на основних популярних операційних системах: Windows, Mac OS, Linux. Тому за допомогою ASP.NET Core ми можемо створювати крос-платформні додатки. І хоча Windows як середовище розробки та розгортання все ще домінує, ми наразі не обмежені цією операційною системою. Тобто ми можемо запускати веб-додатки не тільки в Windows, але і в Linux та Mac OS. А для розгортання веб-програми можна використовувати традиційний сервер IIS або Kestrel. Завдяки модульній рамковості всі компоненти, необхідні для веб-програми, можна завантажувати як окремі модулі через Nuget Package Manager. Крім того, на відміну від попередніх версій платформи, вам не потрібно використовувати каталог System.Web.dll.

ASP.NET Core включає в себе фреймворк MVC, який об'єднує функціональність MVC, Web API і Web Pages. У попередніх версії платформи дані технології реалізувалися окремо і тому містили багато дублюючої функціональності. Зараз же вони об'єднані в одну програмну модель ASP.NET

Core MVC. А Web Forms повністю пішли в минуле. Фреймворк ASP.NET Core довзволяє:

- розгортання в хмарі або локально;
- запуск як на .NET Core так і .NET Framework.

ASP.NET Core - це редизайн ASP.NET 4.x з архітектурними змінами, які призводять до більш компактною і модульній структурі. ASP.NET Core надає наступні переваги:

- має єдину історію створення веб-інтерфейсу і веб-API;
- зручний у тестуванні;
- інструмент Razor Pages робить розробку орієнтованих на сторінки сценаріїв простішим і продуктивним;

  blazor дозволяє використовувати C # в браузері разом з JavaScript;

- спільне використання логіки на стороні сервера і на стороні клієнта, написаної на .NET;
- можливість розробки і запуску на Windows, MacOS і Linux;
- відкритий вихідний код;
- інтеграція сучасних клієнтських середовищ і робочих процесів розробки;
- готова до роботи в хмарі система конфігурації на основі середовища;
- вбудована ін'єкція залежностей;
- легкий, високопродуктивний і модульний конвеєр HTTP-запитів;
- можливість розміщення на IIS, Nginx, Apache, Docker або самостійного розміщення на своєму власному процесі;
- інструмент, який спрощує сучасну веб-розробку.

Entity Framework - це спеціальна об'єктно-орієнтована технологія, заснована на .NET framework для роботи з даними. Якщо традиційні інструменти ADO.NET дозволяють створювати з'єднання, команди та інші об'єкти для взаємодії з базами даних, то Entity Framework є більш високим рівнем абстракції, що дозволяє абстрагуватися від самої бази даних і працювати з даними незалежно від типу сховища. . Якщо на фізичному рівні ми працюємо з таблицями, індексами, первинними і зовнішніми ключами, то на

концептуальному рівні, що пропонує нам Entity Framework, ми вже працюємо з об'єктами.

Перша версія Entity Framework - 1.0 була випущена ще в 2008 році і представлена дуже обмеженою функціональністю, базовою підтримкою ORM (об'єктно-реляційне відображення - відображення даних про реальні об'єкти) і єдиним підходом до взаємодії з базою даних - Database First. З виходом версії 4.0

у 2010 році багато чого змінилося - з тих пір Entity Framework став рекомендованою технологією доступу до даних, а нові можливості взаємодії з підходами Model First і Code First були введені в рамки.

Додаткові поліпшення функціональності відбулися з виходом версії 5.0 в

2012 році. Нарешті, в 2013 році було випущено Entity Framework 6.0, з можливістю асинхронного доступу до даних.

Основна концепція Entity Framework є поняття сутності або entity. Сутність являє собою набір даних, пов'язаних з певним об'єктом. Тому ця технологія передбачає роботу не з таблицями, а з об'єктами та їх множинами.

Будь-яка сутність, як і будь-який об'єкт з реального світу, має ряд властивостей. Наприклад, якщо сутність описує людину, то можна виділити такі властивості, як ім'я, прізвище, висота, вік, вага. Властивості не обов'язково являють собою прості дані типу int, але можуть також представляти більш складні структури даних. І кожен об'єкт може мати одну або кілька властивостей, які будуть відрізняти цю сутність від інших і однозначно визначатиме цю сутність. Ці властивості називаються ключами.

У цьому випадку об'єднання можуть бути з'єднані асоціативністю "один до багатьох", "один-до-одного" та "багато-до-багатьох", як і в реальній базі даних, спілкування відбувається через зовнішні ключі.

Відмінною особливістю Entity Framework є використання запитів LINQ для вибірки даних з бази даних. За допомогою LINQ ми можемо не тільки витягти

певні рядки, які зберігають об'єкти з бази даних, але й отримувати об'єкти, пов'язані різними асоціативними зв'язками.

Іншою ключовою концепцією є модель даних суб'єкта. Ця модель порівнює класи об'єктів з реальними таблицями в базі даних.

ASP.NET Core характеризується розширюваністю. Фреймворк побудований з набору щодо незалежних компонентів. І ми можемо або використовувати вбудовану реалізацію цих компонентів, або розширити їх за допомогою механізму спадкування, або зовсім створити і застосовувати свої компоненти зі своїм функціоналом [11].

Також було спрощено управління залежностями і конфігурація проекту. Фреймворк тепер має свій легкий контейнер для впровадження залежностей, і більше немає необхідності застосовувати сторонні контейнери, такі як Autofac, Ninject. Хоча при бажанні їх також можна продовжувати використовувати.

В якості інструментарію розробки ми можемо використовувати останні випуски Visual Studio, починаючи з версії Visual Studio 2015. Крім того, ми можемо створювати додатки в середовищі Visual Studio Code, яка є крос-платформної і може працювати як на Windows, так і на Mac OS X і Linux.

Для обробки запитів тепер використовується новий конвеєр HTTP, який заснований на компонентах Katana і специфікації OWIN. А його модульність дозволяє легко додати свої власні компоненти.

Якщо підсумувати, то можна виділити наступні ключові відмінності ASP.NET Core від попередніх версій ASP.NET:

- Новий легкий і модульний конвеєр HTTP-запитів
- Можливість розгортати додаток як на IIS, так і в рамках свого власного процесу
- Використання платформи .NET Core і її функціональності
- Поширення пакетів платформи через NuGet
- Інтегрована підтримка для створення та використання пакетів NuGet
- Єдиний стек веб-розробки, що поєднує Web UI і Web API
- Конфігурація для спрощеного використання в хмарі

- Вбудована підтримка для впровадження залежностей
- Можливість розширення
- Кросплатформеність: можливість розробки і розгортання додатків ASP.NET на Windows, Mac і Linux

3.4 Середовище розробки Visual Studio

Microsoft Visual Studio - лінійка продуктів компанії Microsoft, що включають інтегроване середовище розробки програмного забезпечення і ряд інших інструментальних засобів. Дані продукти дозволяють розробляти як консольні додатки, так і додатки з графічним інтерфейсом, в тому числі з підтримкою технології Windows Forms, а також веб-сайти, веб-додатки, веб-служби як в рідному, так і в керованому кодах для всіх платформ, підтримуваних Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework і Silverlight.

Visual Studio включає редактор вихідного коду з підтримкою технології IntelliSense та можливістю повторного кодування коду. Вбудований налагоджувач може працювати як відладчик вихідного коду та відладчик рівня машини. Інші вбудовані інструменти включають редактор форм для спрощення створення графічного інтерфейсу програми, веб-редактора, дизайнера класів та дизайнера схем бази даних. Visual Studio дозволяє створювати та підключати сторонні додатки (розширення) для розширення функціональності майже на кожному рівні, включаючи додавання підтримки для систем управління вихідним кодом (таких як Subversion і Visual SourceSafe), додаючи нові набори інструментів (наприклад, редагування та візуальний дизайн) код у керованих мовах програмування Тема) або інструменти для інших аспектів процесу розробки програмного забезпечення (наприклад, Team Explorer для роботи з Team Foundation Server).

3.5 Засіб розробки Angular

Angular - це платформа та основа для побудови односторінкових клієнтських додатків за допомогою HTML та TypeScript. Кутовий пишеться в TypeScript. Він реалізує основну та додаткову функціональність як набір бібліотек TypeScript, які ви імпортуєте у свої програми.

Кутова архітектура програми базується на певних основних концепціях. Основними будівельними блоками є NgModules, які слугують основою для складання деталей. NgModules збирає відповідний код у функціональні пакети; Кутова програма визначається набором NgModules. У програмі завжди є щонайменше один кореневий модуль, який дозволяє завантажувати і, як правило, має багато інших модулів.

Компоненти визначають подання даних, що представляє собою набір елементів екрану, які вибираються та змінюються відповідно до кута та відповідно до даних вашої програми.Компоненти використовують сервіси, які надають певні функціональні можливості, безпосередньо не пов'язані з переглядами. Постачальники послуг можуть бути введені в компоненти як залежності, що робить ваш код модульним, багаторазовим та ефективним.

І компоненти, і сервіси - це просто уроки, з якими дизайнери вказують їх тип та надають метадані, що показують Angular, як ними користуватися.

Метадані класу компонентів пов'язують його з шаблоном, який визначає подання. Шаблон поєднує звичайний HTML з настановами щодо кута та розмітки, які дозволяють Angular змінювати HTML перед його відображенням.

Метадані для класу обслуговування надають інформацію, яку Angular потребує, щоб зробити її доступною для компонентів через введення залежності (DI).

Компоненти програми зазвичай визначають багато представлень, розташованих ієрархічно. Angular надає послугу маршрутизатора, щоб допомогти вам визначити шляхи навігації серед представлень. Маршрутизатор забезпечує складні навігаційні можливості в браузері.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Реалізація веб додатку для системи обліку надзвичайних ситуацій техногенного характеру включає в себе проектування архітектури додатку, створення моделей бази даних, реалізацію серверної частини включаючи налагодження механізму розгортання, створення веб частини, тестування.

4.1 Проектування архітектури додатку

Проектуючи архітектуру веб додатки, було необхідно забезпечити додатком зручну підтримку, легку розширюваність, використання загальноприйнятих принципів і парадигм програмування. Було вибрано .NET Core в силу його кросплатформенності та багатого функціоналу.

Зокрема, в основі програми був застосований шаблон проектування MVVM

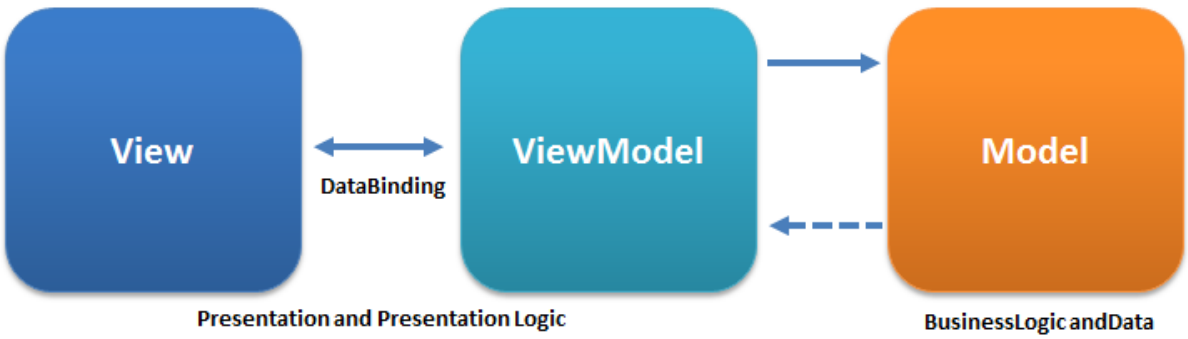


Рисунок 4.1 – Схема роботи MVVM.

Він дозволяє відокремити логіку додатки від візуальної частини (подання). Даний патерн є архітектурним, тобто він задає загальну архітектуру програми.

З огляду на ймовірність розширення веб додатки був також використаний принцип інверсії залежностей, реалізацію якого .NET Core вже включає. Суть методу полягає в двох ключових правилах:

- Модулі верхнього рівня не повинні залежати від модулів нижнього рівня. Обидва повинні залежати від абстракції.
- Абстракції не повинні залежати від деталей. Деталі повинні залежати від абстракцій.

Для авторизації використовується ASP.NET Core Identity:

- Це API, який підтримує функцію входу в інтерфейс користувача (UI).
- Керує користувачами, паролями, даними профілю, ролями, маркерами, підтвердженням електронної пошти тощо.

Користувачі можуть створити обліковий запис із інформацією про вхід, що зберігається в Identity, або вони можуть використовувати зовнішнього постачальника входу.

Ідентифікація налаштована за допомогою бази даних SQL Server, що розміщується в сховищі Azure.

Для зручної маніпуляції з таблицями використовується Entity Framework Core, він являє собою об'єктно-орієнтовану, легковажну і расширяемую технологію від компанії Microsoft для доступу до даних. EF Core є ORM-інструментом (object-relational mapping - відображення даних на реальні об'єкти). Тобто EF Core дозволяє працювати базами даних, але є більш високий рівень абстракції: EF Core дозволяє абстрагуватися від самої бази даних і її таблиць і працювати з даними незалежно від типу сховища. Якщо на фізичному рівні ми оперуємо таблицями, індексами, первинними і зовнішніми ключами, але на концептуальному рівні, який нам пропонує Entity Framework, ми вже працюємо з об'єктами.

4.2 Use-Case діаграма

Розроблена система має відповідати наступній use-case діаграмі (рис 4.2)

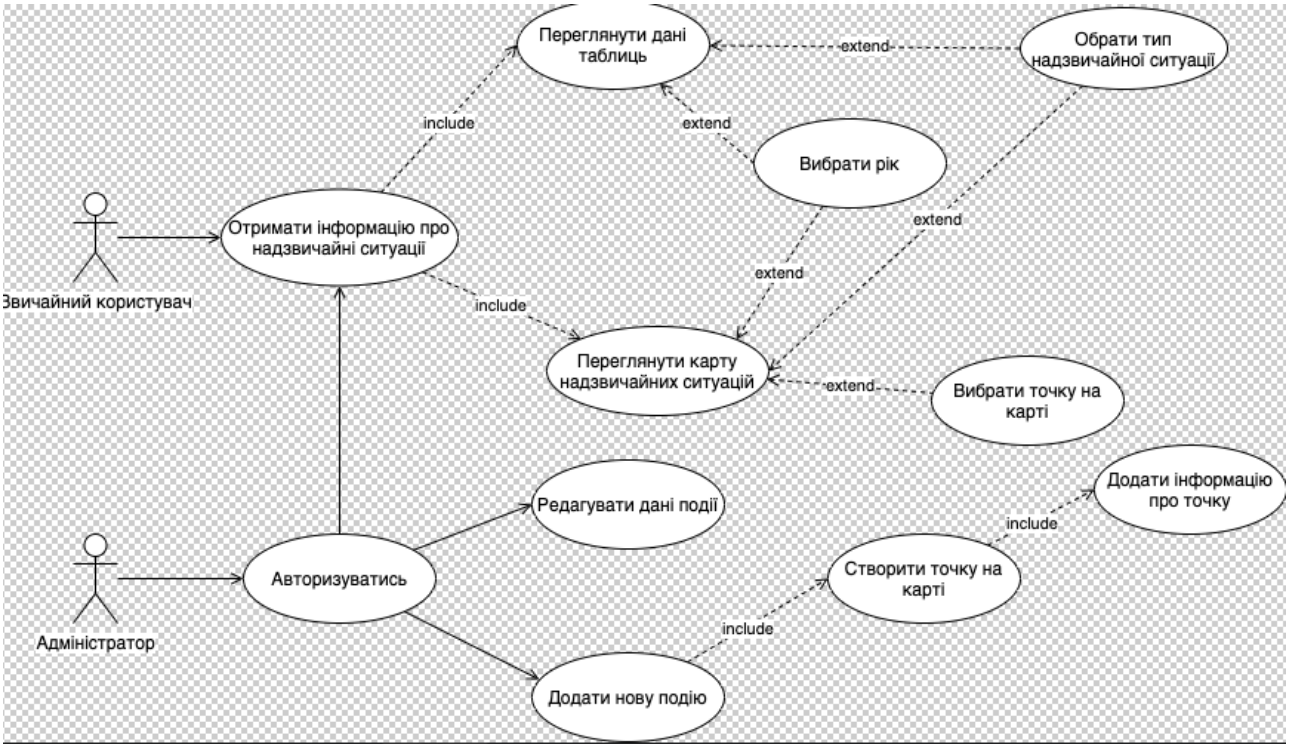


Рисунок 4.2 – Схема бази даних.

4.3 Концептуальна модель бази даних. Створення моделей.

База даних системи складається з трьох таблиць що описують сутності, необхідні для вичерпаного предствлення інформації щодо катастроф різного характеру включаючи техногенні. В базі також присутні таблиці, що надаються системою .NET Core Identity для зберігання інформації про користувачів. База створює єдиний інформаційний простір для зберігання та отримання доступу до даних.

Для створення додатку було вирішено використати model first підхід. Суть даного підходу полягає в тому, що спочатку робиться модель, а потім по ній створюється база даних. Маючи загальне представлення про сутності предметної області були сформовані окремі відповідні моделі (представлені у вигляді класів), що згодом за допомогою механізму міграції були трансформовані в таблиці БД (рис 4.3).

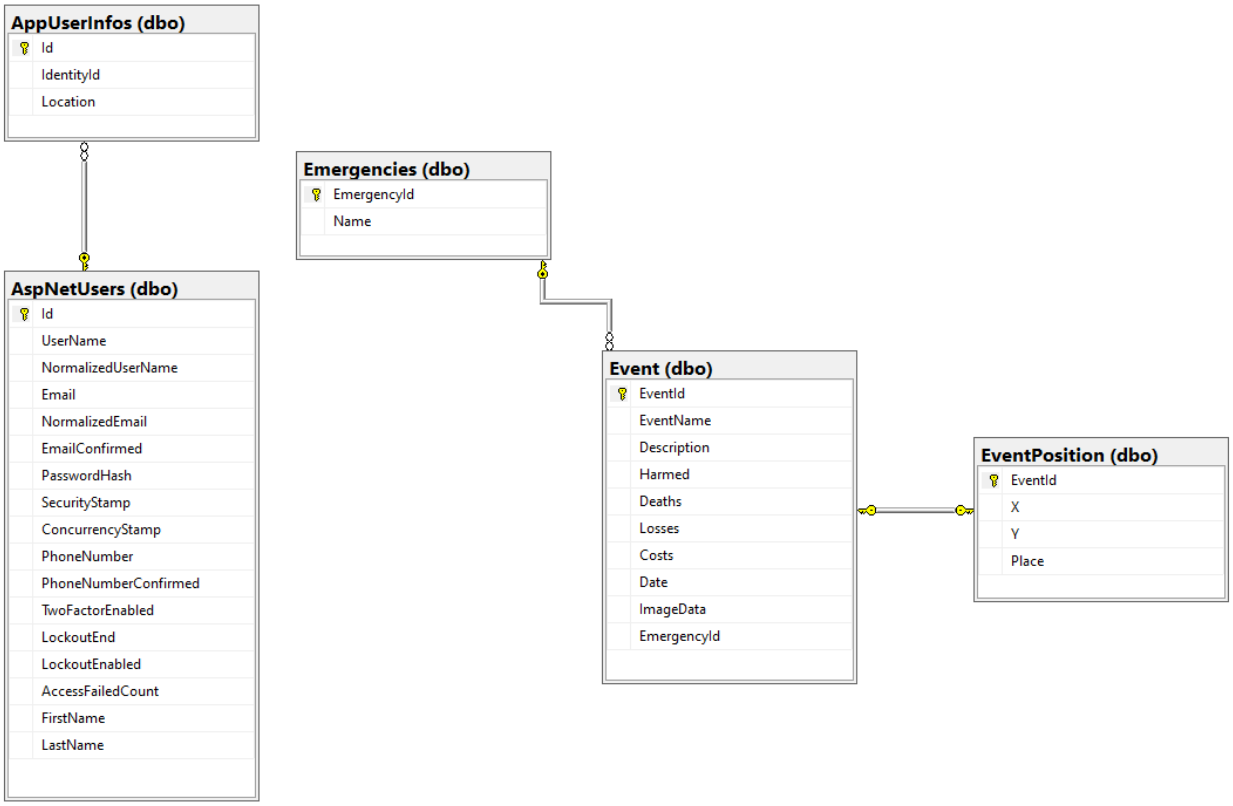


Рисунок 4.3 – Схема бази даних.

У базі даних присутні класичні типи зв’язків реляційної бази даних, такі як «Один до багатьох», а саме відношення dbo.Emergency до dbo.Event, в якому одному типу характеру надзвичайної ситуації може відповідати декілька надзвичайних ситуацій цього типу. Також присутній тип «Один до одного» у випадку з відношенням dbo.Event та dbo.EventPosition, в якому dbo.EventPosition виступає в ролі довідника та містить інформацію про місце катастрофи.

База даних розміщується в Azure SQL Database

База даних SQL Azure - це повністю керована платформа як послуга (PaaS). Двигун бази даних обробляє більшість функцій управління базами даних, такі як оновлення, виправлення, резервне копіювання та відстеження без користування. База даних SQL Azure завжди працює в останній стабільній версії двигуна баз даних SQL Server з доступністю 99,99%. Вбудовані можливості PaaS у базі даних SQL Azure дозволяють зосередитись на управлінні та оптимізації баз даних.

За допомогою бази даних Azure SQL було досягнуто високодоступний і високопродуктивний рівень зберігання даних для додатку у Azure (рис 4.4).

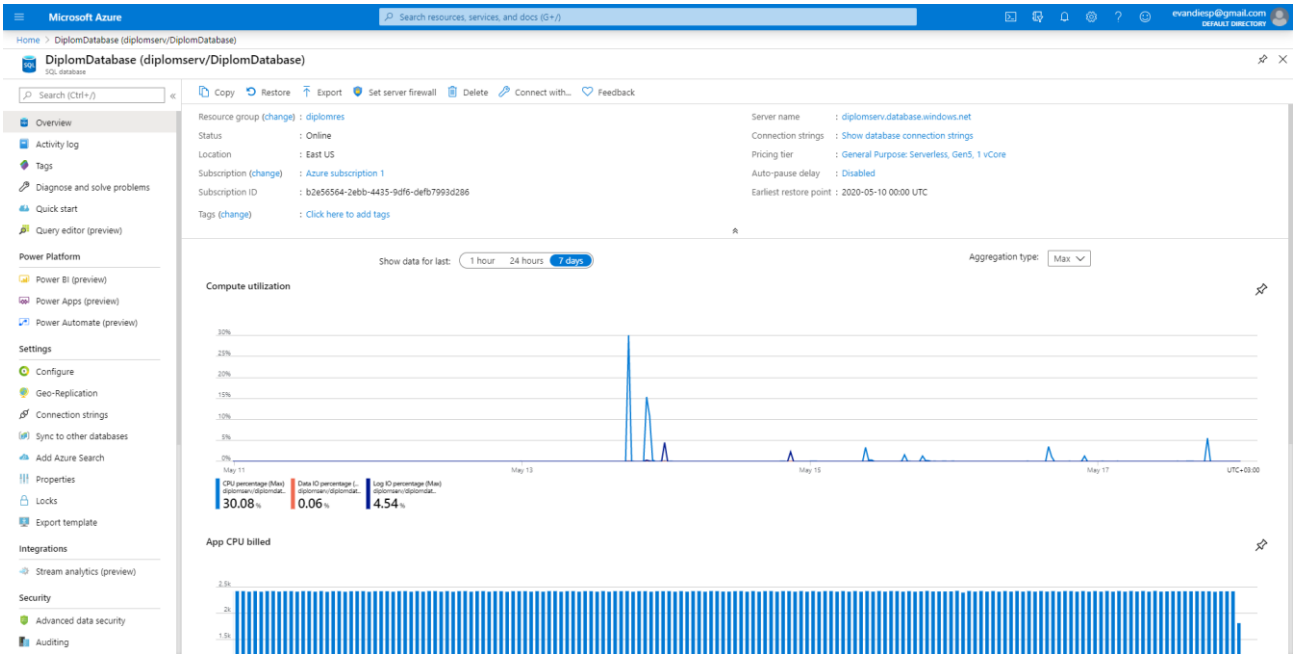


Рисунок 4.4 – Адміністрування Azure Database.

.4.4 Реалізація серверної частини

Серверна частина додатку, створений на .NET Core 3.0, відповідає стандартам розробки веб додатків та включає в себе ряд бажаних підходів як модульне програмування, дотримання принципів SOLID, та інші.

Система авторизації реалізована за допомогою JWT.

JWT (або JSON Web Token) являє собою веб-стандарт, який визначає спосіб передачі даних про користувача в форматі JSON в зашифрованому вигляді. Даний підхід залишає необхідність користувачу кожного разу надсилати пароль для доступу до карти, чи окремих даних системи обліку техногенних катостроф.

Сервер аутентификации проверяет данные пользователей и генерирует для них JWT. Веб-приложение аутентифицируют пользователя по полученному JWT. Приложение поддерживает необходимую шифрацию данных пользователей.

5. РОБОТА КОРИСТУВАЧА З СИСТЕМОЮ

Веб додаток розміщується за посиланням https://diplom20200410104007.azurewebsites.net/map

В разі локального запуску необхідні наступні вимоги: .Net Core 3.0, Angular 7, SQL Server 2016

Після попадання на головну сторінку додатку (рис 5.1) користувач бачить інтерактивну карту України, елементи управління, маркери на мапі.

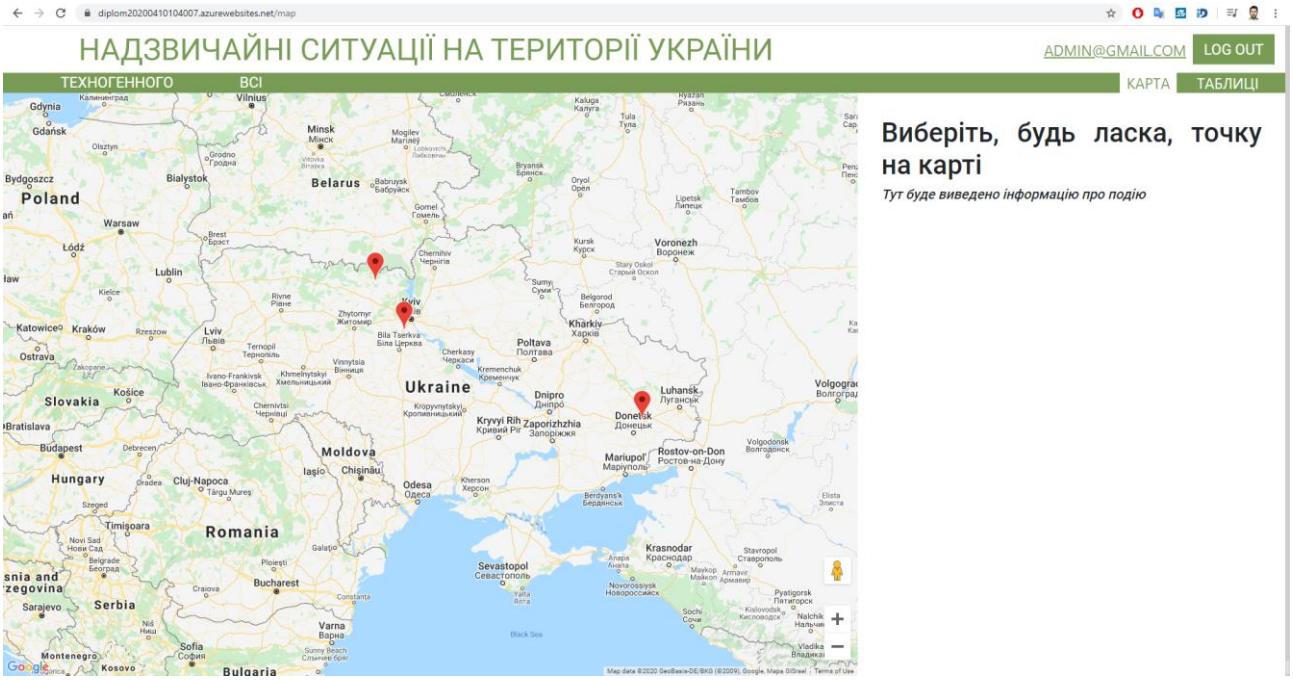


Рисунок 5.1 – Головна сторінка.

Інтерактивна карта містить маркери надзвичайних ситуацій (НС). Користувач має можливість натиснути на один із них для отримання більш детальної інформації (рис 5.2).

Додаток містить панель, що відображає фотографії з місць відповідних катастроф різних характерів.

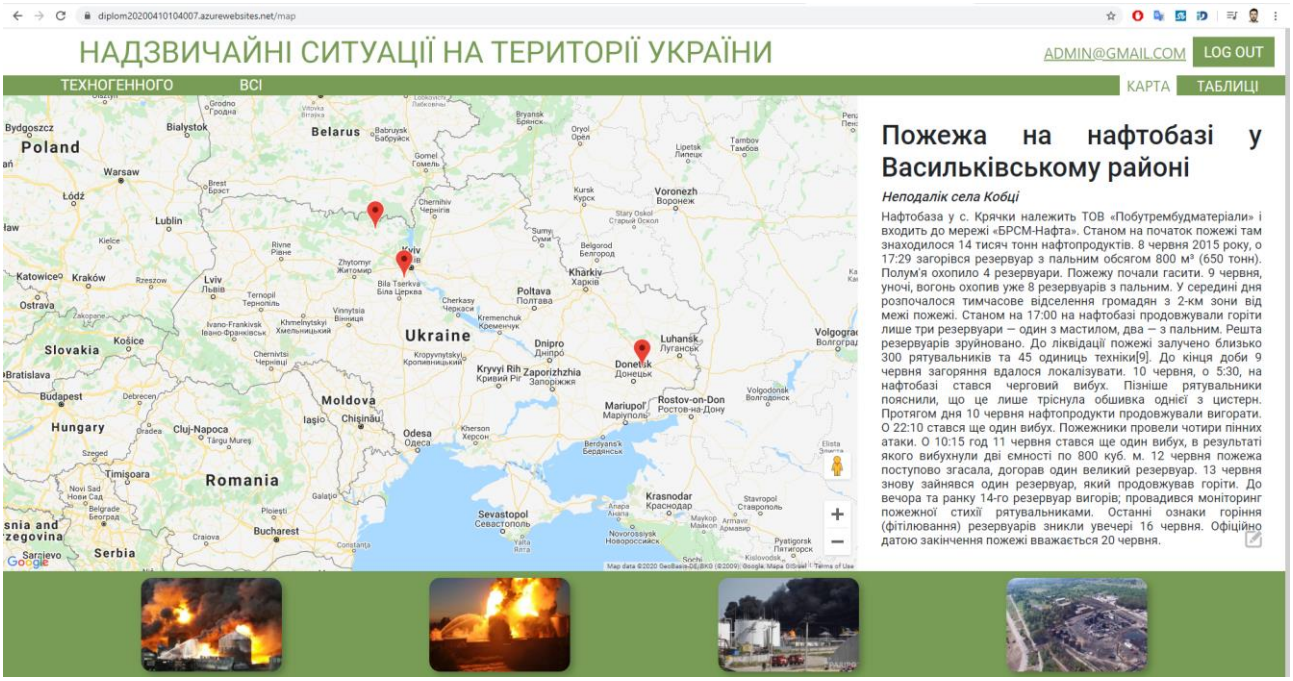


Рисунок 5.2 – Інформація про НС.

Додаток підтримує групування по типу НС (рис 5.3) та по року НС (рис 5.4)

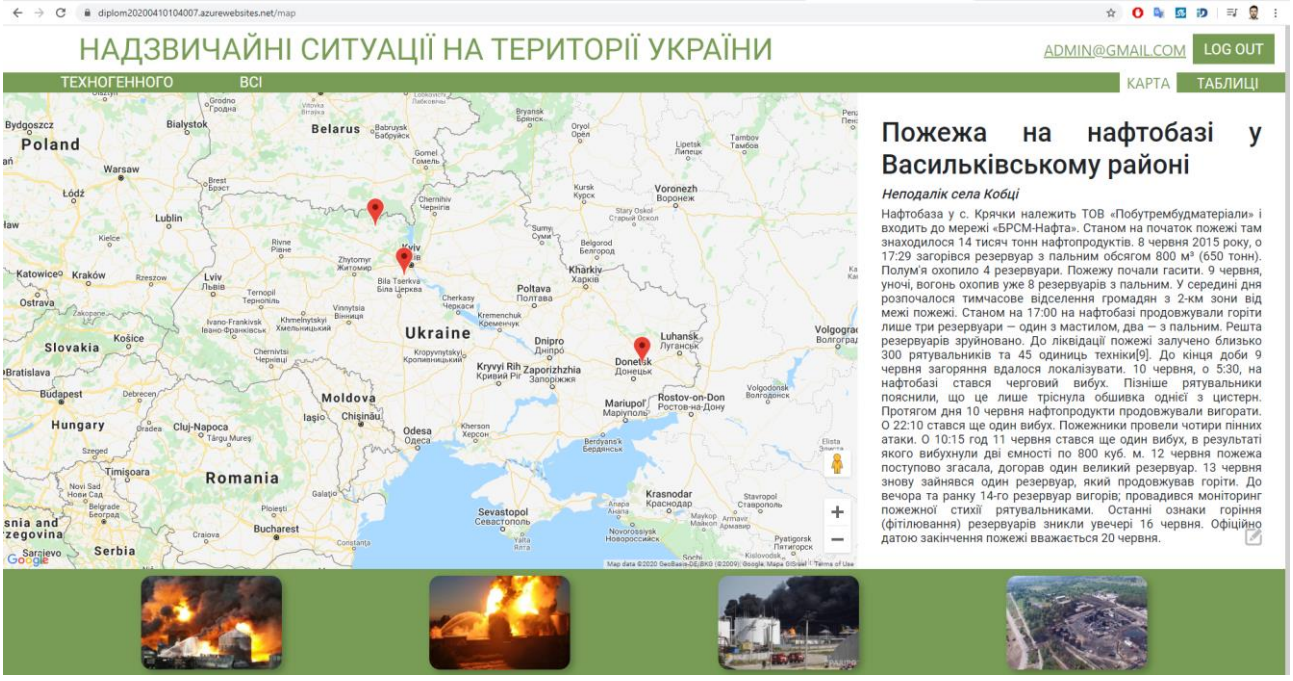


Рисунок 5.3 – Вибір характеру НС.

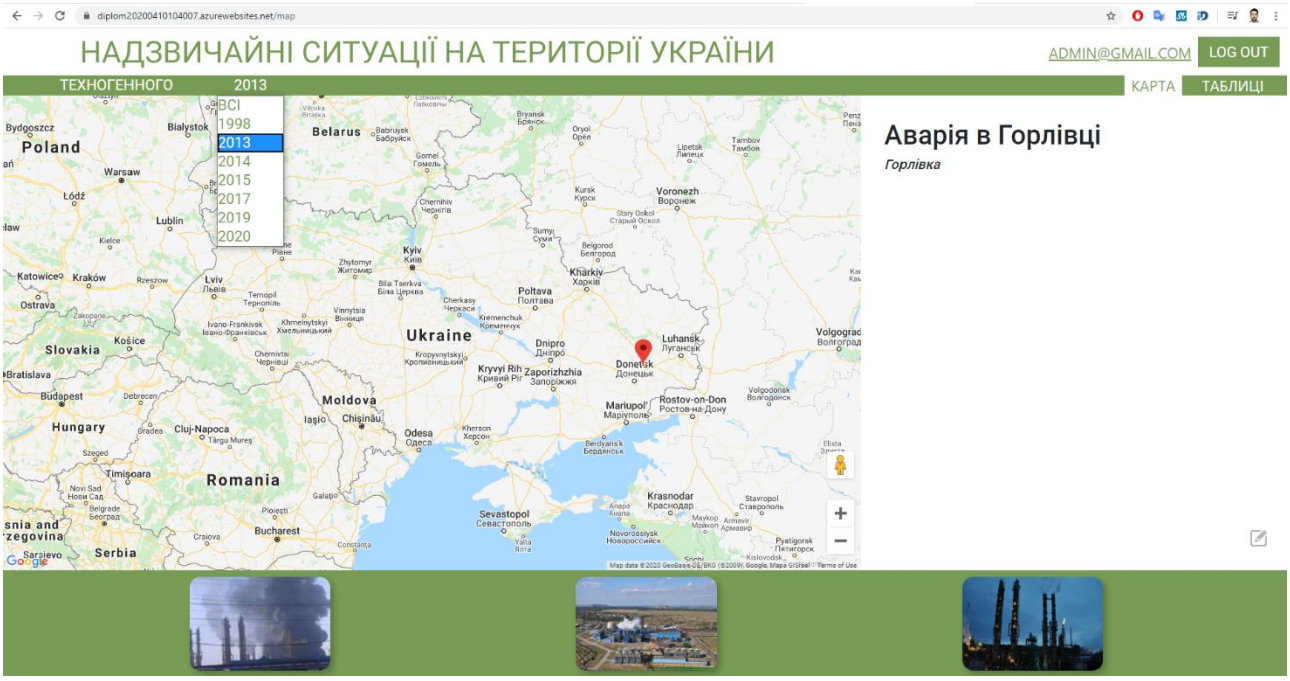


Рисунок 5.4 – Вибір року НС.

Адміністратору пропонується пройти етап авторизації на відповідній сторінці системи (рис 5.5), що дає змогу створювати нову НС (рис 5.6). Необхідні для заповнення поля: Назва НС, Місце НС, Опис ситуації, Дата, Характер НС (рис 5.7).

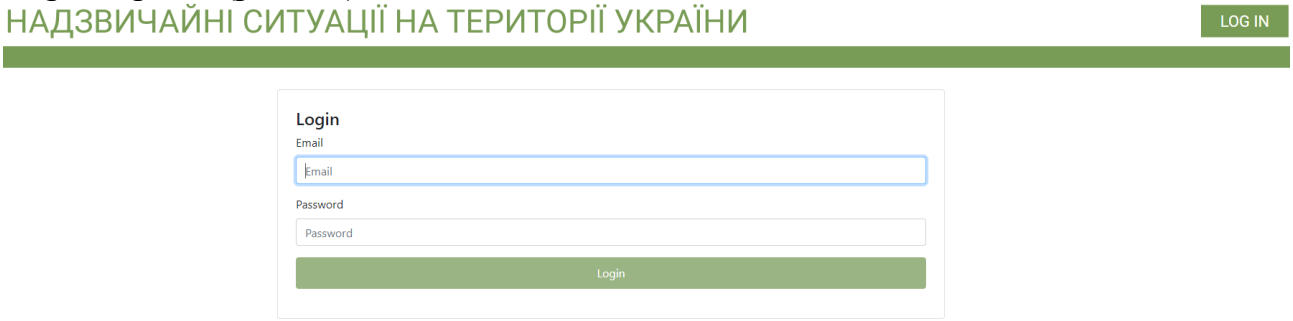


Рисунок 5.5 – Робота адміністратора системи.

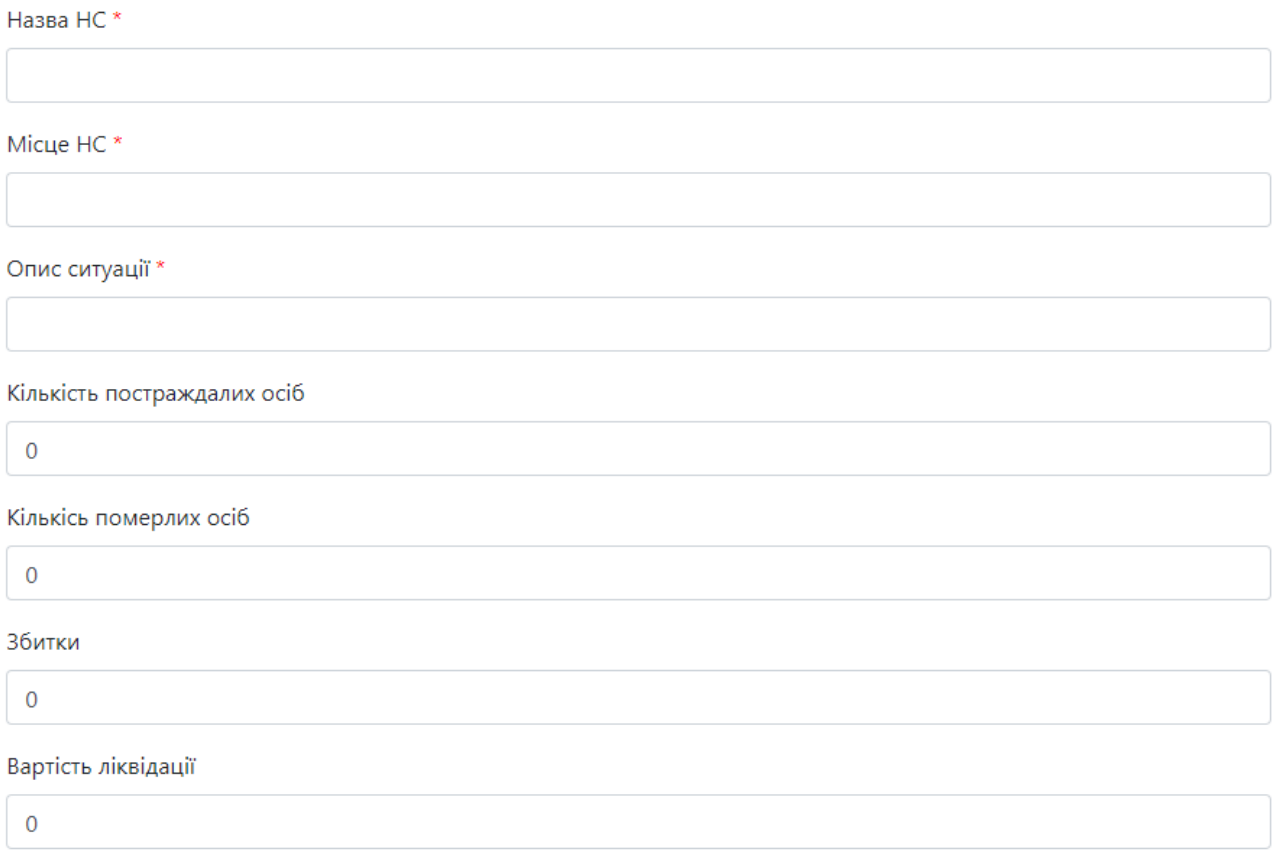


Рисунок 5.6 – Додавання нової НС.

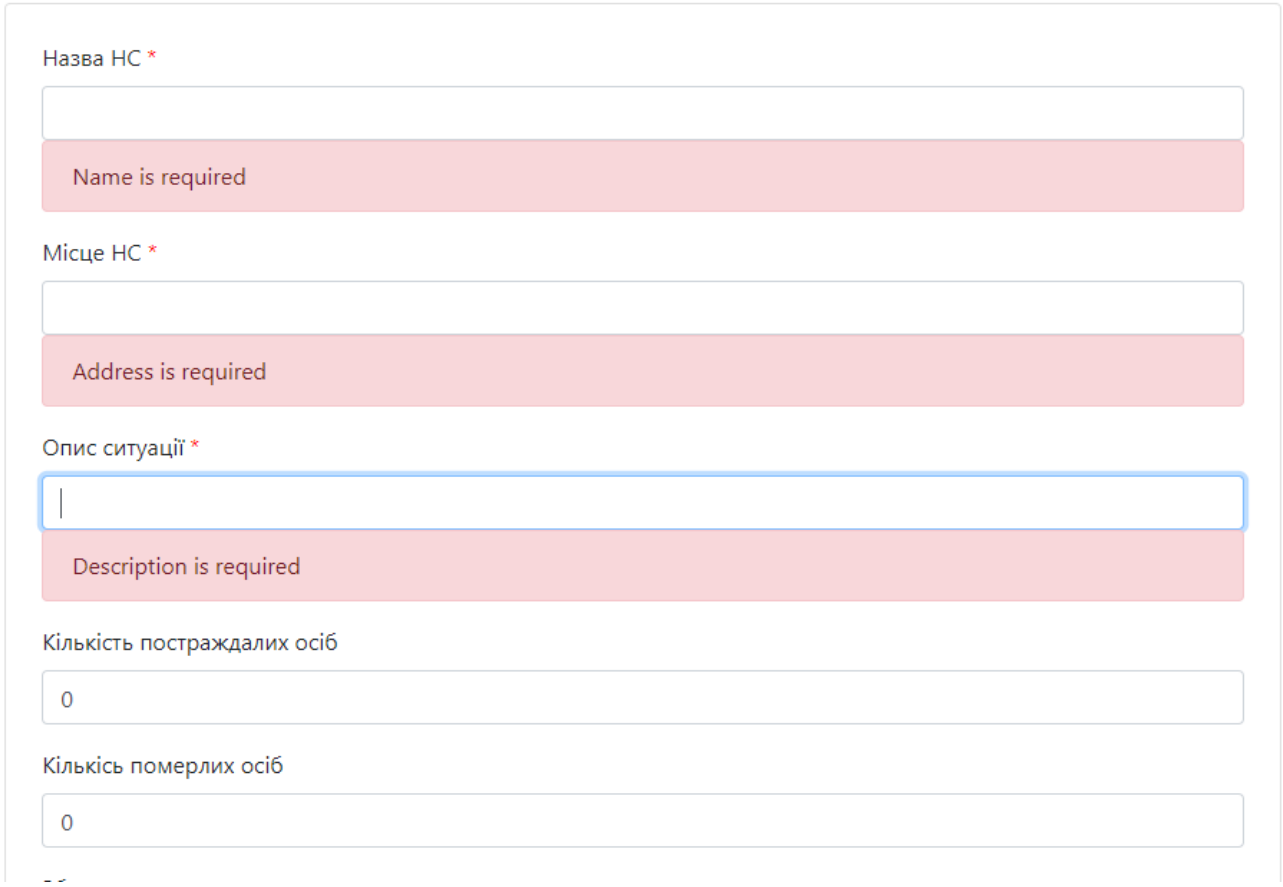


Рисунок 5.7 – Обов’язкові поля.

Під час додавання нової НС, користувач має змогу інтерактивно вибрати місце на мапі та завантажити необхідну кількість фотографій для вичерпного відображення ілюстрацій щодо катастрофи (рис 5.8).

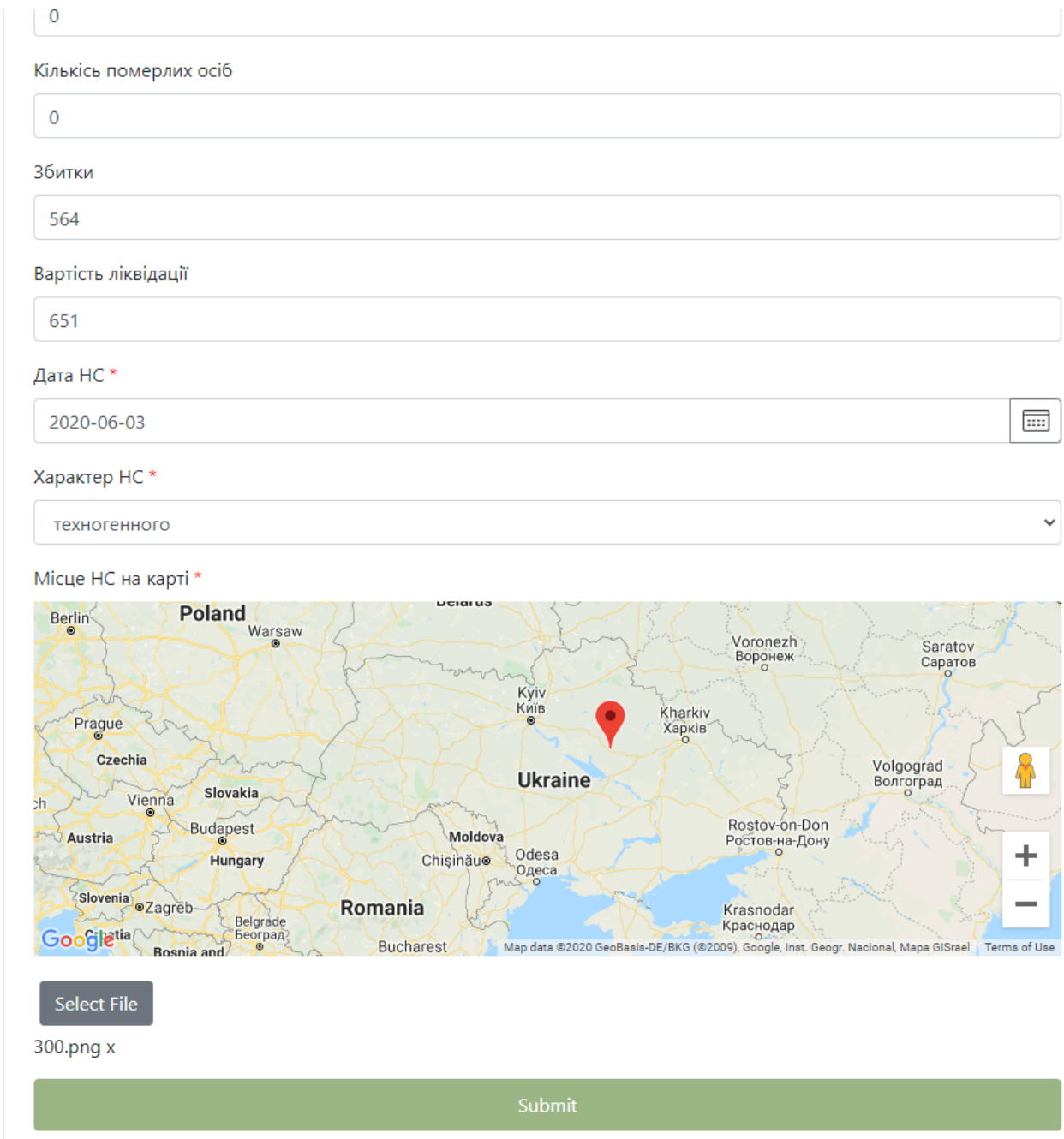


Рисунок 5.8 – Вибір місця та завантаження фотографій.

Вкладка «Таблиці» (рис 5.9) містить загальну інформацію про всі надзвичайні ситуації. Присутня таблиця, що інформує загальну інформацію про всі надзвичайні ситуації в системі обліку НС. Існує можливість групування по характеру НС та року.

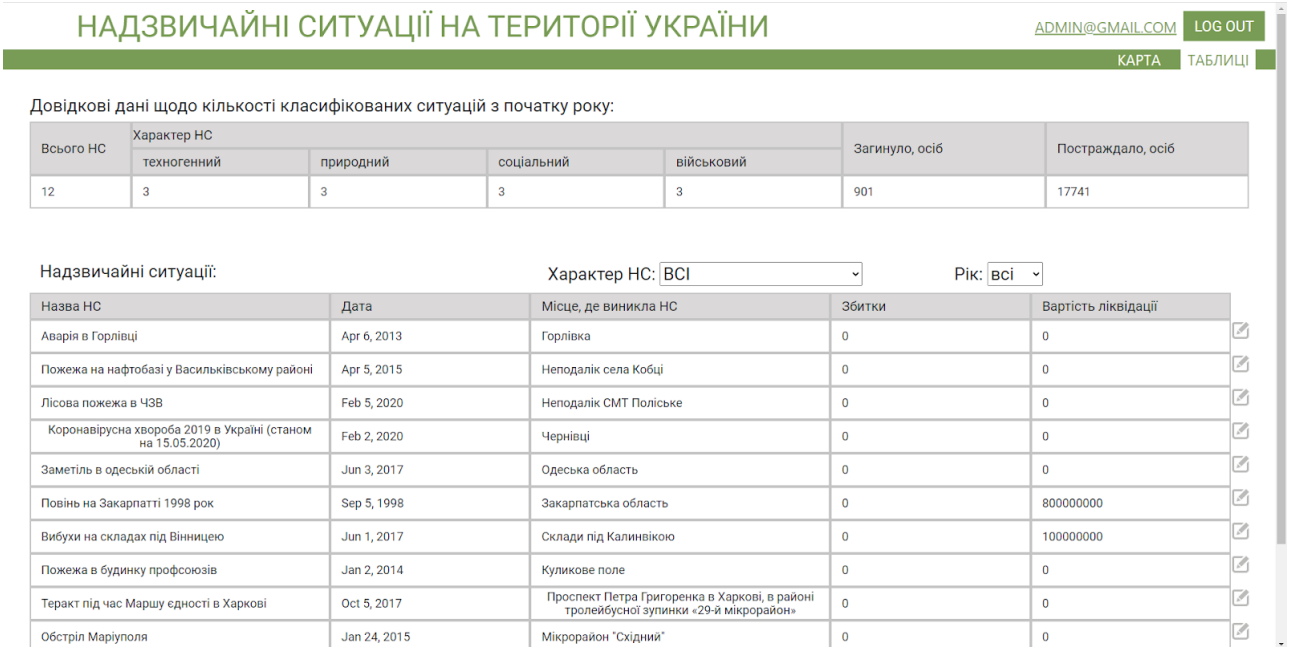
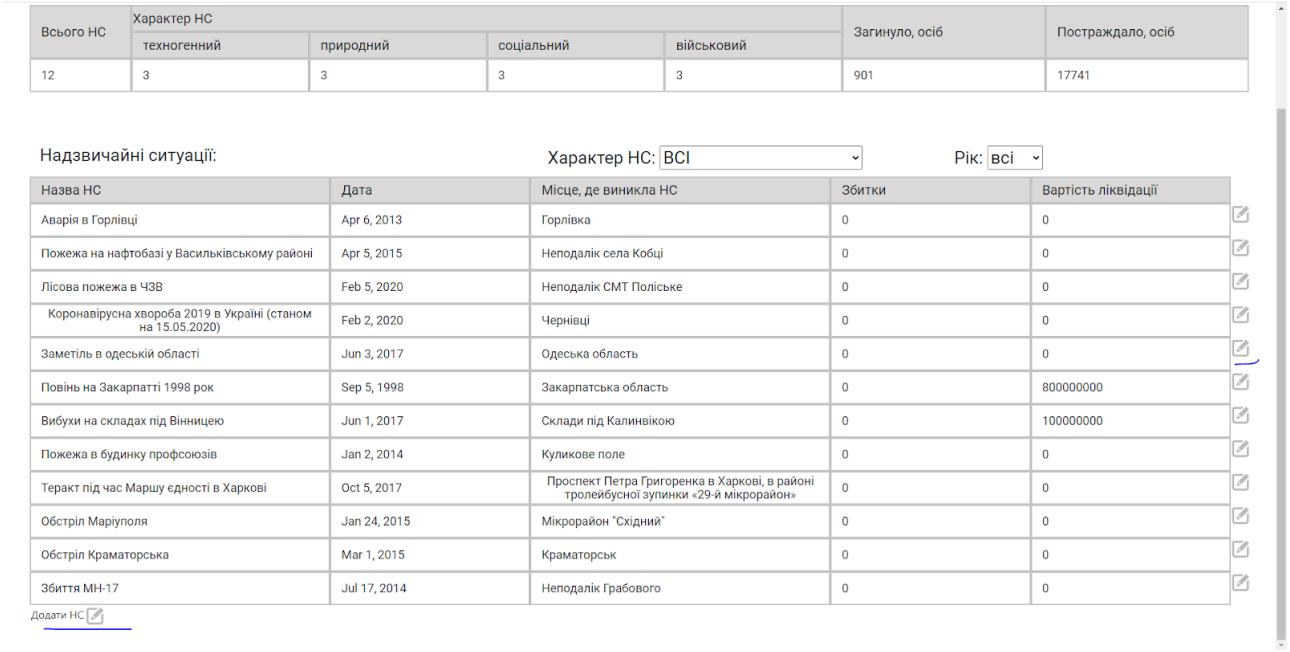


Рисунок 5.9 – Вкладка «Таблиці».

На вкладці «Таблиці» користувач має змогу перейти на вікно редагування надзвичайної ситуації або створити нову натискаючи відповідні кнопки (рис 5.10).



Всього НС	Характер НС				Загинуло, осіб	Постраждало, осіб
	техногенний	природний	соціальний	військовий		
12	3	3	3	3	901	17741

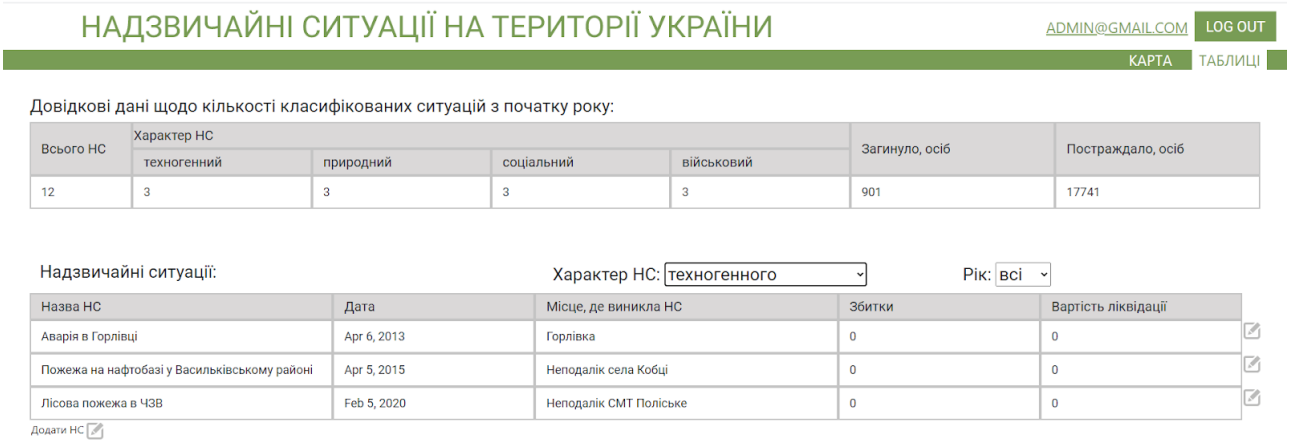
Надзвичайні ситуації: Характер НС: ВСІ Рік: всі

Назва НС	Дата	Місце, де виникла НС	Збитки	Вартість ліквідації
Аварія в Горлівці	Apr 6, 2013	Горлівка	0	0
Пожежа на нафтобазі у Васильківському районі	Apr 5, 2015	Неподалік села Кобці	0	0
Лісова пожежа в ЧЗВ	Feb 5, 2020	Неподалік СМТ Поліське	0	0
Коронавірусна хвороба 2019 в Україні (станом на 15.05.2020)	Feb 2, 2020	Чернівці	0	0
Заметіль в одеській області	Jun 3, 2017	Одеська область	0	0
Повінь на Закарпатті 1998 рок	Sep 5, 1998	Закарпатська область	0	800000000
Вибухи на складах під Вінницею	Jun 1, 2017	Склади під Калинівкою	0	100000000
Пожежа в будинку профсоюзів	Jan 2, 2014	Куликове поле	0	0
Теракт під час Маршу єдності в Харкові	Oct 5, 2017	Проспект Петра Григоренка в Харкові, в районі тролейбусної зупинки «29-й мікрорайон»	0	0
Обстріл Маріуполя	Jan 24, 2015	Мікрорайон "Східний"	0	0
Обстріл Краматорська	Mar 1, 2015	Краматорськ	0	0
Збиття МН-17	Jul 17, 2014	Неподалік Грабового	0	0

Додати НС

Рисунок 5.10 – Кнопки управління.

Існує можливість сортування значень таблиці за характером НС (рис 5.11)



НАДЗВИЧАЙНІ СИТУАЦІЇ НА ТЕРИТОРІЇ УКРАЇНИ ADMIN@GMAIL.COM LOG OUT

КАРТА ТАБЛИЦІ

Довідкові дані щодо кількості класифікованих ситуацій з початку року:

Всього НС	Характер НС				Загинуло, осіб	Постраждало, осіб
	техногенний	природний	соціальний	військовий		
12	3	3	3	3	901	17741

Надзвичайні ситуації: Характер НС: техногенного Рік: всі

Назва НС	Дата	Місце, де виникла НС	Збитки	Вартість ліквідації
Аварія в Горлівці	Apr 6, 2013	Горлівка	0	0
Пожежа на нафтобазі у Васильківському районі	Apr 5, 2015	Неподалік села Кобці	0	0
Лісова пожежа в ЧЗВ	Feb 5, 2020	Неподалік СМТ Поліське	0	0

Додати НС

Рисунок 5.11 – Сортування за НС.

Також існує можливість сортування значень таблиці за роком НС (рис 5.12)

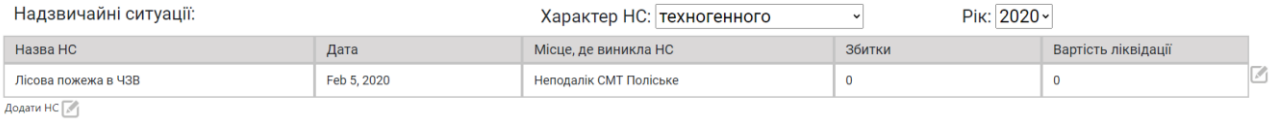


Рисунок 5.12 – Сортування за НС за роком.

ВИСНОВКИ

При виконанні дипломної роботи було досліджена тема техногенних катастроф. Були розширенні знання в створенні та проектуванні веб додатків, а саме покращене знання платформи розробки .NET Core, Angular, навички роботи з системами управління базами даних та хмарними рішеннями від Microsoft Azure. Здобуті практичні навички в розробці системи обліку надзвичайних ситуацій техногенного характеру що в значній мірі полегшує роботу з інформацією стосовно надзвичайних ситуацій на території України. Покращене розуміння роботи клієнт-серверної архітектури.

В результаті була створено очікувана система, що дає змогу вести облік та демонстрацію надзвичайних ситуацій на карті інтерактивним чином включаючи фотографії з місць НС.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 20 років Чорнобильської катастрофи. Погляд у майбутнє: Національна доповідь України [Електронний ресурс] — К.: Атіка, 2006. — 224 с. — Режим доступу: http://www.mns.gov.ua/chornobyl/20_year/03/ n_report_UA.pdf .

2. Стихийные бедствия и техногенные катастрофы. Превентивные меры) // Издательство Альпина Паблишер [Електронний ресурс] — Режим доступу: https://www.amazon.com/Microsoft-C-Language-Specifications-MSDN/dp/0735614482

3. Арнольд В.И. Теория катастроф. – М.: Наука, 1990.

4. Безопасность жизнедеятельности: Учебник для вузов/ С.В.Белов, А.Ф. Козьяков, А.В. Ильницкая. Исправ. и допол.- М.: Высш.шк.; 2006.

5. Безопасность жизнедеятельности. Охрана труда: учеб. пособие для вузов/ А.В. Фролов, Т.Н. Бакаева: Ростов-на-Дону: Феникс, 2005.

6. Безопасность общества и человека в современном мире: Учебное пособие. – СПб.: Политехника, 2005.

7. Белов С.В. Проблемы безопасности при чрезвычайных ситуациях. – М.: 2003.

8. Долин П.А. Ликвидация чрезвычайной ситуации. М., Энергоиздат, 1992.

9. Microsoft C# Language Specifications (MSDN) [Електронний ресурс] — Режим доступу: https://www.amazon.com/Microsoft-C-Language-Specifications-MSDN/dp/0735614482.

10. CLR via C# (4th Edition) [Електронний ресурс] / Jeffrey Richter// ООО Издательство «Питер», 2013— Режим доступу: https://viduus.net/wp-content/uploads/2018/02/Rihter-Dzh.-CLR-via-C.-Programmirovanie-na-platforme-Microsoft-.NET-Framework-4.5-na-yazyke-C-Master-klass-2013.pdf

11. Джеймс Чамберс — ASP.NET Core. Разработка приложений [Електронний ресурс]. — 2018. — Режим доступу: https://bit.ly/2AliYL2.

ДОДАТОК А

Система обліку надзвичайних подій
техногенного характеру (розробка веб-інтерфейсу користувача).

Специфікація

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТМ62202_9Б

Аркушів 1

Київ 2020

Позначення	Найменування	Примітки
Документація		
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ ТМ62202_9Б	Записка.docx	Пояснювальна записка
Компоненти		
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ ТМ62202_9Б 12-1	EventsController.cs BlobStorageContentProvider.cs EmergencyRepository.cs	Основні компоненти
УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ ТМ62202_9Б 13-1	Додаток В.doc	Опис програмного модуля

ДОДАТОК Б

Система обліку надзвичайних подій
техногенного характеру (розробка веб-інтерфейсу користувача).

Текст програми

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТМ62202_9Б 12-1

Аркушів 8

Київ 2020

```
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc;
using System.Threading.Tasks;
using Diplom.DataModels;
using Microsoft.AspNetCore.Http;
using Diplom.Services;
using System;
using Microsoft.AspNetCore.Authorization;
using Diplom.ViewModels;
using System.Text.RegularExpressions;
using Newtonsoft.Json.Linq;
using Newtonsoft.Json;

namespace Diplom.Controllers
{
    [ApiController]
    [Route("api/[controller]")]
    public class EventsController : ControllerBase
    {
        private readonly IEventService _eventService;

        public EventsController(IEventService eventService)
        {
            _eventService = eventService;
        }

        [HttpGet]
        [ProducesResponseType(StatusCodes.Status200OK)]
        public async Task<IEnumerable<Event>> GetAsync()
        {
            return await _eventService.ListAsync();
        }

            [HttpGet("{eventId}")]
        [ProducesResponseType(StatusCodes.Status200OK)]
        public async Task<IEnumerable<Event>> GetAsync(int eventId)
        {
            return await _eventService.ListAsync(eventId);
        }

        [HttpGet("dates/{date}")]
        [ProducesResponseType(StatusCodes.Status200OK)]
        public async Task<IEnumerable<Event>> GetAsync(DateTime date)
        {
            return await _eventService.ListAsync(date);
        }

        [HttpGet("{date}/{emergencyId}")]
        [ProducesResponseType(StatusCodes.Status200OK)]
        public async Task<IEnumerable<Event>> GetAsync(DateTime date, int emergencyId)
        {
            return await _eventService.ListAsync(date, emergencyId);
        }

        [HttpGet("dates")]
        [ProducesResponseType(StatusCodes.Status200OK)]
        public async Task<List<DateTime?>> GetDatesAsync()
        {
            return await _eventService.DatesListAsync();
        }

        [HttpPut("update")]
        [ProducesResponseType(StatusCodes.Status200OK)]
        public async Task<ActionResult> UpdateEventAsync([FromBody] JObject jEventDTO)
        {
            JsonSerializer serializer = new JsonSerializer();
            EventDTO eventDTO = (EventDTO)serializer.Deserialize(new JTokenReader(jEventDTO), typeof(EventDTO));
            try
            {
                var result = await _eventService.UpdateEventAsync(eventDTO);

                if (!result.Success)
                    return BadRequest(result.Message);

            }
            catch (Exception e)
            {
                return BadRequest(e.Message);
            }
```

```
            return Ok();
        }

        // POST: api/Event/add
        [HttpPost("add"), DisableRequestSizeLimit]
        [ProducesResponseType(StatusCodes.Status200OK)]
        public async Task<ActionResult> AddEventAsync([FromBody] JObject jEventDTO)
        {
            JsonSerializer serializer = new JsonSerializer();
            EventDTO eventDTO = (EventDTO)serializer.Deserialize(new JTokenReader(jEventDTO), typeof(EventDTO));
            try
            {
                var result = await _eventService.AddEventAsync(eventDTO);

                if (!result.Success)
                    return BadRequest(result.Message);

            }
            catch (Exception e)
            {
                return BadRequest(e.Message);
            }

            return Ok();
        }

        // POST: api/Event/delete
        [HttpPost("delete"), ActionName("Delete")]
        public async Task<IActionResult> Delete(int eventId)
        {
            var result = await _eventService.DeleteAsync(eventId);

            if (!result.Success)
                return BadRequest(result.Message);

            return Ok();
        }

    }
}
```

```
using Diplom.DataModels;
using Microsoft.Azure.Storage;
using Microsoft.Azure.Storage.Blob;
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Threading.Tasks;

namespace Diplom.Repositories
{
    public class BlobStorageContentProvider : IContentProvider
    {
        private readonly CloudBlobClient _client;

        public BlobStorageContentProvider(string connectionString)
        {
            if (CloudStorageAccount.TryParse(connectionString, out var storageAccount))
            {
                _client = storageAccount.CreateCloudBlobClient();
            }
            else
            {
                throw new AccessViolationException();
            }
        }

        public async Task RemoveByFullPath(List<string> pathList)
        {
            var container = _client.GetContainerReference(Constants.BlobContainerName);

            foreach (var path in pathList)
            {
                var blob = container.GetBlobReference(path);

                var isExist = await blob.ExistsAsync();
```

```
        if (!isExist)
        {
            continue;
        }

        await blob.DeleteIfExistsAsync();
    }
}

public async Task RemoveImages(Event @event)
{
    var container = _client.GetContainerReference(Constants.BlobContainerName);

    if (String.IsNullOrEmpty(@event.ImageData))
    {
        return;
    }

    foreach (var fullPath in @event.ImageData.Split(Constants.Delimiter))
    {
        var blob = container.GetBlobReference(Path.GetFileName(fullPath));

        var isExist = await blob.ExistsAsync();

        if (!isExist)
        {
            continue;
        }

        await blob.DeleteIfExistsAsync();
    }
}

public async Task UploadBlobs(Event @event, List<byte[]> blobs)
{
```

```
            var container = _client.GetContainerReference(Constants.BlobContainerName);

            var paths = new List<string>();

            foreach (var blob in blobs)
            {
                var blobName = Guid.NewGuid().ToString().ToLower() + ".png";

                var uploadBlob = container.GetBlockBlobReference(blobName);

                paths.Add(uploadBlob.Uri.AbsoluteUri);

                using (var stream = new MemoryStream(blob, writable: false))
                {
                    await uploadBlob.UploadFromStreamAsync(stream);
                }
            }

            @event.ImageData = String.Join(Constants.Delimiter, paths);
        }
    }
}
```

```
using Diplom.Data;
using Diplom.DataModels;
using Diplom.ViewModels;
using Microsoft.EntityFrameworkCore;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;

namespace Diplom.Repositories
{
    public class EmergencyRepository : BaseRepository, IEmergencyRepository
    {
        public EmergencyRepository(DiplomDatabaseContext context) : base(context) { }

        public async Task<StatisticViewModel> GetStatisticAsync()
        {
            var all = await _context.Emergencies.Include(e => e.Events).ToListAsync();

            var tech = all.Find(e => e.EmergencyId == Constants.TechnogenicId);
            var natr = all.Find(e => e.EmergencyId == Constants.NaturalId);
            var socl = all.Find(e => e.EmergencyId == Constants.SocialId);
            var mili = all.Find(e => e.EmergencyId == Constants.MilitaryId);

            int technogenicNumber = tech.Events == null ? 0 : tech.Events.Count;
            int naturalNumber = natr.Events == null ? 0 : natr.Events.Count;
            int socialNumber = socl.Events == null ? 0 : socl.Events.Count;
            int militaryNumber = mili.Events == null ? 0 : mili.Events.Count;


            int deathNumber = 0;
            int harmedNumber = 0;

            if (tech.Events != null)
            {
                deathNumber +=  tech.Events.Select(t => t.Deaths).Sum().Value;
                harmedNumber += tech.Events.Select(t => t.Harmed).Sum().Value;
```

```
    }

    if (natr.Events != null)
    {
        deathNumber += natr.Events.Select(t => t.Deaths).Sum().Value;
        harmedNumber += natr.Events.Select(t => t.Harmed).Sum().Value;
    }

    if (socl.Events != null)
    {
        deathNumber += socl.Events.Select(t => t.Deaths).Sum().Value;
        harmedNumber += socl.Events.Select(t => t.Harmed).Sum().Value;
    }

    if (mili.Events != null)
    {
        deathNumber += mili.Events.Select(t => t.Deaths).Sum().Value;
        harmedNumber += mili.Events.Select(t => t.Harmed).Sum().Value;
    }

    var vm = new StatisticViewModel()
    {
        EventsNumber = all.Select(eme => eme.Events.Count).Sum(),
        TechnogenicNumber = technogenicNumber,
        NaturalNumber = naturalNumber,
        SocialNumber = socialNumber,
        MilitaryNumber = militaryNumber,

        DeathsCount = deathNumber,
        HarmedCount = harmedNumber
    };

    return vm;
}

public async Task<IEnumerable<Emergency>> ListAsync()
```

```
        {
            return await _context.Emergencies.Include(e => e.Events).ToListAsync();
        }

        public async Task<IEnumerable<Emergency>> ListAsync(string EmergencyName)
        {
            return await _context.Emergencies.Where(e => e.Name == EmergencyName).Include(e => e.Events).ToListAsync();
        }

        public async Task<IEnumerable<Emergency>> ListAsync(int emergencyId)
        {
            return await _context.Emergencies.Where(e => e.EmergencyId == emergencyId).Include(e => e.Events).ToListAsync();
        }
    }
}
```

ДОДАТОК В

Система обліку надзвичайних подій
техногенного характеру (розробка веб-інтерфейсу користувача).

Опис програми

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТМ62202_9Б 13-1

Аркушів

Київ 2020

АНОТАЦІЯ

Розділ містить опис частини, яка слугує для роботи з данними та картою, що є основними одиницями програмного продукту, та забезпечує поєднання усії інший компонентів для функціонування веб додатку. Вказано принцип роботі з API, який слугує для отримання потрібних даних з серверу та бази даних. Модуль написано мовою програмування C#, з використанням EntityFramework.

ЗМІСТ

ЗАГАЛЬНІ ВІДОМОСТІ

У додатку розглядається один з програмних модулів системи — модуль для роботи з веб-сторінкою з кодом УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТМ62202_9Б 12-1, що міститься у файлі EventsController.cs. Модуль реалізовано за допомогою C# та бібліотек BCL. Модуль призначений для управління системою, яка відповідають за керування даними (API).

ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ

Призначенням модулю для роботи з веб-картою є відображення інформації з серверу та бази даних та безпосереднього прийняття даних введення та обмін інформацією із клієнтським інтерфейсом. Використання такого шаблону дозволяє створювати програмне забезпечення, де інтерфейс і логіка роботи модуля для бази даних та серверу є незалежними компонентами, що дає можливість використовувати його для зменшення навантаження на клієнтську частину системи.

ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Слідкувати за зміною інформації про підсистеми є головним завданням модуля. При запуску системи модуль повертає всю інформацію, яка міститься в базі даних, для відображення даних на користувацькому інтерфейсі. Також модуль оброблює інформацію, надаючи змогу вводити потрібні дані та проводити обчислення та відображати результати у вигляді таблиць та на карті.

ТЕХНІЧНІ ЗАСОБИ ЩО ВИКОРИСТОВУЮТЬСЯ

Модуль розроблено у середовищі розробки Microsoft Visual Studio, що забезпечує набір сервісних функцій та графічний діалог з користувачем, на комп’ютері, має встановлений браузер та підключення до інтернету. Для реалізації клієнтської частини було використано Angular. За допомогою елементів Angular клієнтська частина може з’єднуватись з сервером за допомогою API.

В якості провайдера самої інтерактивної карти було обрано відкритий сервіс GoogleMaps.

ВИКЛИК І ЗАВАНТАЖЕННЯ

Програмний модуль реалізований як окремий клас, який забезпечує існування клієнтської частини та бізнес-логіки окремо від одного, але разом із цим запуск обох компонентів відбувається одночасно.

Для використання цього модуля треба завантажити веб-сторінку, яка зробить потрібні запити та відобразить результат.

ВХІДНІ І ВИХІДНІ ДАНІ

Вхідними даними для модуля є інформація, яку користувач вводить в додатку та місцеположення.

Вихідними даними програмного модуля є відображення збережених даних інтерактивно на веб карті та у вигляді таблиць.

ДОДАТОК Г

Система обліку надзвичайних подій
техногенного характеру (розробка веб-інтерфейсу користувача).

Довідки про впровадження результатів роботи

УКР.НТУУ”КПІ”_ТЕФ_АПЕПС_ТМ62202_9Б 14-1

Аркушів 1

Київ 2020